EFFICIENT ALGORITHMS FOR SPEECH RECOGNITION OF CANTONESE

BY

LAI WAI MING

(黎慧明)

A MASTER THESIS SUBMITTED IN PARTIAL

FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF PHILOSOPHY

IN

THE DEPARTMENT OF ELECTRONICS

THE CHINESE UNIVERSITY OF HONG KONG

HONG KONG

MAY, 1987.

ACKNOWLEDGEMENTS

I acknowledge gratefully the valuable guidance and encouragement given by my supervisor, Dr. P. C. Ching. Thanks are also due to Professor Y. T. Chan* for many stimulating discussions and suggestions. In addition, I would like to express my appreciation to those who have helped in recording.

* Department of Electrical Engineering, Royal Military College of Canada, Kingston, Ontario, Canada.

ABSTRACT

Automatic speech recognition has received a great deal of attention in the past decade and a wide variety of isolated-word recognition systems have been used in many applications. However, the temporal aligning techniques employed in most template-based recognizers require a large amount of computation. Speech recognizers based on hidden Markov models, on the other hand, have less computation but more complication parameter estimation procedures for model generation. Large storage and computational requirements, together with complex system configurations, have limited the possibility of a high speed, low-cost implementation of these recognition algorithms, even for a small vocabulary.

A novel speech recognition scheme is proposed which is essentially template-based, but avoids the necessity for temporal alignment. The method uses the energy-time profiles (ETP) of a word at different frequency bands as the parameter for recognition. Each of the band-pass filtered signals is divided into a fixed number of segments regardless of the duration of the input token, and the energy of each of these segments is recorded and normalized to form the ETP matrix. Recognition is then effected by matching the ETP of a test word against those of the reference templates. To reduce the matching time further, a zero-crossing count front end classifies a word as to be either fricative initial or voiced initial. Thus a word, after classification, will only be matched against one of the two groups of references. Various methods of forming the reference templates using the iterative clustering technique have been examined and evaluated.

The algorithm is especially suitable for monosyllabic languages such as Mandarin and Cantonese, and can be implemented very easily on a microcomputer. A real-time recognition system may be achieved by apportioning a large part of the pre-processing task to external hardware. The system is used for both speaker-dependent and speaker-independent isolated-word recognition of the ten Cantonese digits. An accuracy of 99 percent was obtained for speaker-dependent recognition. For speaker-independent recognition, an average of 93-95 and 83-87 percent were obtained for trained and untrained speakers respectively.

CONTENT

CHAPTER 1 INTRODUCTION

In the past twenty years, speech processing, especially speech recognition, has gained immense attention throughout the world and has become a major research topic both at academic institutions as well as numerous research institutes in industry. The recent advances of high speed digital computers and the rapid growth in the area of artificial intelligence of robotics has stimulated the development of speech recognition to a great extent. Primarily, the main objective of speech recognition is to establish efficient communication interface with machines by human voice instead of keyboard or paper tape reader. The advantages of speech input are being that high input speed can be obtained and no training is needed. In addition, parallel processing in conjunction with actions of the hands and feet or with visual perception of information is possible and last but not the least, simple and economical input sensors are available.

The development of sophisticated speech recognition algorithms have shown tremendous potential for widespread use in the future [1]. The basic task of a speech recognition system is either to recognize the entire spoken utterance exactly, or else to 'understand' the spoken utterance. The concept of understanding rather than recognizing the utterance is of most important for systems which deal with fairly large vocabulary continuous speech input, whereas the concept of exact recognition is of most important for limited vocabulary, small speaker population, isolated word systems.

For a fixed vocabulary size, the levels of complexity involved in speech recognition are of course dependent on the system capability.

Continuous speech, speaker independent recognition is perhaps the most difficult, and isolated-word, speaker-dependent the easiest. The reasons being that continuous speech recognition requires first isolating the speech into distinct words before recognition; and speaker-independent recognition can only be achieved through using many reference templates for each word, obtained from many different speakers during training.

Speech recognition is usually accomplished by some form of pattern matching. Input utterance is first acoustically analyzed and speech characteristics are extracted to give a parametric representation of the spoken word. During recognition, these parameters are compared with prestored reference patterns obtained from a training session. If the input is found to match closely with one of the reference pattern, it will be recognized as the word associated with this pattern. It has been shown that human hearing system is much like a highly tuned spectrum analyzer with the capability of determining the specific amplitudes across the audio spectrum. Hence, two kinds of information, a speech spectrum envelope and the characteristics of an exciting source, are normally used as the parameters for speech recognition. However, for robust measurements, the parameters including zero-crossing rate, energy at different frequency bands, cepstrum coefficients, and linear predictive coding (LPC) coefficients are usually employed. The linear prediction analysis [2] has been proven to be the most efficient method in extracting the spectrum envelop characteristics of a speech segment. This is done by computing the coefficients of a linear filter that models the human vocal tract transfer function. Many systems

developed in laboratory environment employing this technique have been shown to achieve very high recognition accuracies [3 - 6]. But the heavy computational requirement of this method has made real-time implementation of the system very difficult, if not impossible. Consequently, alternative recognition techniques that require less computations have been widely studied as well. One of the favourable approaches is the filter bank analysis in which the input signal is first passed through a series of bandpass filters. The rectified outputs of these filters form a vector representation of the speech spectrum and is used as a parameter set for recogniton. The performance of this kind of systems highly depends on the choice of the filter bank. Yet another method is to separate the excitation and impulse response of the vocal tract by the homomorphic deconvolution technique, and the so-called cepstrum coefficients are extracted as the parameters for comparisons. Despite the high recognition rate that might be achieved by these systems, one common drawback is that they all have to employ the dynamic time warping algorithm to align speech signals with different durations. This warping process performs the search of the best path that matches the test token and reference template nonlinearly. Obviously, this involves excessive amount of calculation and thus slows down the recognition speed considerably.

More recently, recogniton systems that based on the speech production mechanism and classification of speech sounds have been investigated extensively [7 - 10]. These are referred to as the phonetically-based recognition systems. Speech utterance can be denoted by a sequence of phonetic labels which characterize its

acoustic and linguistic features. In the reference set, each word within the vocabulary is normally represented by its phonetic structure as well as the acoustic parameter vectors. When testing with an unknown input, the speech is identified both by its phonetic and parameter pattern. Since the variations of the parameter sets along the time axis have already been accomplished by the phonetic labels, temporal alignment is not needed. This means that dynamic time warping can be alleviated and hence the computational load is reduced. However, these recognizers always employ complicated decision schemes which lead to very complex system models. In addition, these systems are not suitable for recognizing monosyllabic languages owing to their lacking of phonetical variations in individual words.

In the late 1960's, the basic theory of Markov chains was first applied to the area of speech processing and had attracted much attention. But it is only in the past decade that it has been applied explicitly to problems in speech recognition. Continued refinements in the theory and implementation of Markov modelling techniques have lead to the development of several successful speech recognizers [11, 12]. Instead of comparing acoustic features, the hidden Markov model (HMM) for each word in the vocabulary is generated as the reference, and a probability score for each word HMM is computed on an unknown input such that the word corresponding to the model with the highest probability score is chosen as the recognized word. Although it has been shown that an HMM recognition system requires approximately 17 times less computation than an equivalent recognizer using LPC coding and dynamic time warping [11], they demand complicated parameter

estimation procedures for model generation. Due to the complexity of the algorithm, all modelling have to be done on big mainframe computers with expensive special-purpose hardware which makes low-cost microcomputer implementation impossible [13].

It is noted that all the recognition techniques discussed so far are not suitable for the development of a real-time system utilizing simple and inexpensive hardware. In this project, we intend to propose a small vocabulary, isolated word speech recognition scheme that is capable of providing a reasonably high performance and also possible for low cost real-time implementation. The system is designed for monosyllabic tonal languages, specifically for Cantonese. Currently, most of the existing recognizers have been developed for polysyllabic languages such as English or Japanese. However, almost all dialects spoken by the Chinese people, including Mandarin and Cantonese, are monosyllabic languages. It is believed that the distinctive energy-time profiles of monosyllabic languages might provide simple means for recognition. Therefore, our objective is to design an efficient recognition technique catered for Cantonese which has a large population of speakers in Southern China and in Hong Kong.

The recognition scheme is essentially template-based, but does not involve dynamic programming for time alignment. The energy-time profile (ETP) of a word at different frequency bands is extracted as the parameter for recognition. The filter bank is chosen in such a way that the formant frequencies can be explicitly obtained from the ETP vectors. Instead of manipulating the parameters on a fixed duration basis, the input signal is evenly divided into a fixed number

of segments and measurements are made within each frame. As a result, a linear compression or extrapolation is effectively applied on the signal. Since the dimension of the feature vectors is now identical for all tokens, time warping process is not required during matching. It has been found that this kind of linear temporal alignment is only applicable to monosyllabic languages because for polysyllabic sound, energy profiles at different frequency bands might not be able to preserve their spectral shapes if projected linearly on the time axis. In order to reduce the matching time further, a zero-crossing count front end is employed to accomplish a voiced/fricative initial classification. The traditional Euclidean distance measure with the addition of nonlinear normalization is used for template matching. The normalized absolute difference is also introduced as an alternative for distance measure which involves less computations since no squaring operation is needed. A detailed description of the system configuration and the method of distance normalization will be given in Chapter 4.

In the reference set, each word is represented by multiple templates which are created based on the clustering technique with many different speakers. The modified K-means clustering algorithm proposed by Wilpon et al. [14] is applied with some modifications to fit our system requirement. Four different sets of observations have been tried for clustering, namely the whole ETP matrices, the ETP vectors of individual frequency band, the energy vector at each time frame and the individual segment energy. Several experiments were conducted to evaluate the performance of these four methods for both speaker-dependent and speaker-independent recognition. In addition,

comparisons are made to examine the difference in recognition score between no classification and that with classification, as well as between using squared distance measures and absolute distance measures. For speaker-independent recognition, a semi-open test and an open test have been performed separately. In Chapter 5, the clustering algorithm for template generation and the various tests will be discussed in detail. The results of these tests are summarized in table form so as to give a clearer illustration. Comments and discussions of the results are included in Chapter 6. Conclusion together with recommendations for further studies will also be given in this chapter.

It is worth to note that the major advantage of the proposed system is in its simplicity and it can be implemented on a microcomputer very easily. Furthermore, as a large part of the pre-processing task including endpoint detection, segmentation and bandpass filtering can be realized by standard hardware, a real-time recognition system may be achieved without much difficulty.

CHAPTER 2 SPEECH RECOGNITION USING DYNAMIC FEATURES OF SPECTRUM

2.1 Spectral-Based Speech Recognition Systems

Human perception of speech sounds involves a series of operations of the auditory system [15]. The sound waves enter the ear where they are converted from acoustic to mechanical vibrations by the eardrum and the ossicles. A travelling wave is set up in the cochlea, and a preliminary analysis based principally on frequency is performed. The signal is transformed into a set of neural discharges by the hair cells, and additional frequency selectivity is carried out. The resulting pattern of neural activity is transmitted to the auditory cortex. Although the analysis is mostly in terms of frequency, the temporal pattern of the waves is retained as much as possible by synchronous discharges of the neurones.

A number of theories have evolved in the past century trying to explain how the ear converts the incoming sound waves to physiological signals for the brain. One of the earliest theories, known as the "resonance" theory, considered the basilar membrane to be a series of independently tuned resonators. These resonators performed a Fourier analysis and transmitted the intensity of each frequency components to the brain as the strength of the corresponding nerve discharges. Based on the concept of the "resonance" theory, the "place" theory was developed which found that the capability of frequency discrimination comes, instead of from the tuned resonators, actually from the place of vibration within the cochlea. The frequency related signals are transmitted to the brain in a parallel-serial combination through the

acoustic nerves and the signals are then analysed and recognized. In other words, the information of discrete frequency components is carried by their corresponding nerve fibres to the brain in a parallel manner, whereas the amplitude or loudness is represented by the number of impulses along each frequency dependent nerve fibre. That means, our hearing system operates just like a highly tuned spectrum analyzer which performs a short-time spectral analysis over the audio spectrum.

In 1952, Delattre et al. revealed that the frequencies of the lowest two formants are the most important features for recognizing a vowel [15]. They synthesized vowels containing one, two and more formants, and noticed that two formants were necessary in order to produce a complete set of vowels. Experiments had also been performed to investigate the perception of semivowels and consonants. It was found that the duration of the formant transitions and the initial frequency of the second formant transition were the main cues for the recognition of semivowels. The stop consonants or plosives, on the other hand, should be distinguished by the formant transition and the frequency of the noise burst at the moment of release. Nasal sounds have a distinct characteristic of possessing a strong formant at about 200Hz and relatively weak formants in the rest of their spectra, and could be separated from one another by the formant transitions due to the closing and opening of the oral tract. Finally, for discriminating the fricatives, the high frequency spectrum and the transition of the formants in the adjacent voiced part were needed.

It is still an unknown that how exactly the human auditory system analyses and interprets incoming speech signals. However, the spectral properties, especially the formant frequencies, are certainly

one of the major features for the recognition of speech sounds. Indeed, much efforts on machine recognition have been spent in order to extract a unique set of spectral properties to distinguish different speech sounds. Several parameter sets, such as outputs of filter banks, spectrum envelope, cepstrum, autocorrelation function, partial correlation coefficients, and linear predictive coding (LPC) coefficients, have been evaluated in various recognition systems. In the mean time, LPC is the most popular approach while cepstrum and filter banks are still areas of interest.

The philosophy of linear prediction is intimately related to the basic discrete-time production model. It was shown that speech can be modelled as the output of a linear, time-varying system excited by either quasi-periodic pulses (during voiced sounds), or random noise (during unvoiced sounds) [16]. The linear prediction method provides a robust, reliable and accurate technique for estimating the parameters that characterize the linear, time-varying system and from which the fundamental speech parameters such as the pitch, formant, spectra and vocal tract area function can be derived. The method of linear prediction has been used extensively in the field of machine recognition of human speech [3 - 6]. One typical example is Itakura's system in which every word was represented by a pattern of LPC coefficients obtained from equal-length time segments. During recognition, the pattern of an input utterance was compared with the references using a similarity measure based on the logarithm of the ratio of prediction residuals [4]. Since the durations of different utterances varied even for the same speaker repeating the same word, the number of time segments differed from one utterance to another. A

time-warping function has been employed to map the input time axis into the reference time axis nonlinearly. This function was searched recursively by the algorithm of dynamic programming so as to achieve a minimum distance. This technique of dynamic time warping will be discussed in detail later in this chapter. The vocabulary of the system consisted of 200 Japanese geographical names and the recognition rate was found to be around 97.3 percent. The system was also tested with the vocabulary consisting English alphabet and digits. The recognition rate dropped to 88.6 percent for the same speaker. The majority of confusions occurred between pairs having the same vowel and very little difference in the consonant part.

Several other performance criteria for distance measures were also developed for template matching purposes [17]. A common strategy of these methods is that they are all spectral in nature and energy pattern of the speech signals are not considered. More recently, Brown and Rabiner proposed that the speech energy pattern information should be added to the LPC feature space [5]. A distance function has been defined for comparing the energy patterns which is given by the logarithm of the ratio of normalized frame energy of the test utterance and the references. The total distance between two time frames of different tokens is obtained by summing the log likelihood distance of the LPC vectors and the weighted energy distance. It was shown that the probability of error with both energy and LPC as parameters for comparisons is smaller than that with LPC alone.

Despite the high performance that might be achieved using LPC, the computations involved are substantial which make real-time implementation of the speech recognition systems rather difficult. An

alternative approach is to use a series of filter banks to extract the spectral features of the speech signals. In this method, the speech is first passed through a series of bandpass filters which divide the frequency spectrum of interest into various bands and the number might vary from 3 to 32. These filters are usually continuous over the frequency spectrum and the composite response of the overall filter bank is essentially flat. This assures that equal weighting is given to all frequency components. The frequency spacing of the filter bank can be determined in a number of ways [18], and a fairly standard technique is to divide the frequency spectrum uniformly. The energy of the output of each bandpass filter is measured and is encoded by a logarithmic transformation. The time variation of these energy vectors defines a pattern for the speech. During recognition, dynamic time warping is again adopted for matching the test pattern with the reference pattern. The distance between two time frames can be calculated by summing the squared difference of each corresponding pair of elements. This kind of distance measure is usually referred to as the Euclidean distance.

White and Neely [6] have shown that an LPC-based recognition system and a filter bank recognizer could achieve approximately identical recognition accuracy on some standard word vocabularies. In their studies, the LPC-based system used a 10kHz sampling rate and fourteen coefficients for every 12.8ms of speech segment while the filter bank system consisted of either six one octave filters or 20 one-third octave filters covering the frequency spectrum from about 100Hz to 10kHz. When using a 20-channel filter bank, the average accuracy obtained was about 98.8 percent. If the number of channels

was reduced to 6, the score dropped to 96 percent. The LPC based system, on the other hand, yielded an accuracy of 97 percent.

Besides LPC and filter bank approach, there exists a less popular method which makes use of the cepstrum of speech for recognition. The fundamental idea of cepstrum is based on the homomorphic deconvolution of a signal. Furui has built a recognition system [19] using this technique and the parameters employed include the time functions of the cepstrum coefficients, the log energy, the cepstrum regression coefficients, as well as the energy regression coefficients. The regression coefficients were defined in such a way to represent the slope of the time function of each parameter. Time alignment was achieved by dynamic time warping. Experiments have been carried out to observe the effects of using different combinations of the four parameters on a vocabulary of 100 Japanese city names. The error rates were 6.2, 3.1, and 2.4 percent when using cepstrum coefficients, a combination of cepstrum and their regression coefficients, and a combination of cepstrum and regression coefficients for cepstrum and energy contours respectively. These results give a clear indication of the significance of the dynamic spectral features (in the form of regression coefficients) in speech recognition. The combination of instantaneous (cepstrum) and dynamic features is effective in reducing the frequency of large individual speaker error rate and in reducing unlikely confusions from the perspective of human speech perception.

2.2 Dynamic Time Warping

The recognition systems that have been discussed so far in this

chapter are template-based and all of them utilized the, so called, dynamic time warping (DTW) technique for temporal alignment. If the feature vectors for the speech signal are computed on a short time basis then time alignment must be performed. It is because the duration of human speech varies tremendously, and this speaking rate variation causes nonlinear fluctuation of speech pattern in the time domain. Linear normalization technique was first introduced in which timing differences between speech patterns were eliminated by linear compression or extrapolation of the speech signals [20]. This has been shown to be capable of improving th recognition accuracy significantly but it is insufficient when dealing with highly complicated fluctuation nonlinearities such as polysyllabic sounds. The time warping technique using dynamic programming was later introduced around 1970s as a way of establishing optimum nonlinear matching between input and reference speech patterns [21]. In this algorithm, timing differences between two speech patterns are eliminated by warping the time axis of one so that maximum coincidence is attained with the other. Figure 2.1 illustrates the function of time alignment between a test pattern $T(m) = \{T(1), T(2), \ldots, T(M)\}$ and a reference pattern $R(n) = \{R(1), R(2), \ldots, R(N)\}$ which are expressed as a sequence of feature vectors. The dynamic programming equation may be in the form

$$D_{m,n} = d_{m,n} + \text{Min}(D_{m,n-1}, D_{m-1,n}, D_{m-1,n-1})$$

where $d_{m,n}$ is defined to be the distance between the test utterance at time slice m and the reference utterance at time slice n. $D_{m,n}$ is the total distance between the two utterances from their beginnings up to and including times m and n. The operation Min(a, b, c) selects the

smallest number from the set of numbers a, b, and c. For a more thorough discussion, see [20] and [22].

Since the warping function approximates the properties of actual time-axis fluctuation, several restrictions are imposed to preserve linguistically essential structures such as continuity and monotonicity in a speech pattern. It has been widely accepted that dynamic time warping has played an important role in the area of speech recognition. Indeed, White and Neely [6] have shown that for polysyllabic isolated-word recognition, significant improvements in recognition performance could be obtained when using time warping.

It is worth noting that DTW requires enormous computational efforts though it enhances the recognition performance. The distances of all possible paths have to be evaluated before the best fit warping function can be chosen. Hence, the computational load is increased by three to four times as compared with the linear time alignment method, and thus exaggerates the recognition time. This drawback inevitably hinders real-time realization of these systems. Consequently, researchers resort to other methods that may avoid the warping operation. One of these methods, instead of representing the time pattern of the speech by the feature vectors, describes the acoustic waveform by a sequence of phonetic labels. Input signal is thus recognized by its phonetic structure and/or the associated feature parameters on a statistical basis. Speech recognition systems using this technique will be discussed in Chapter 3.

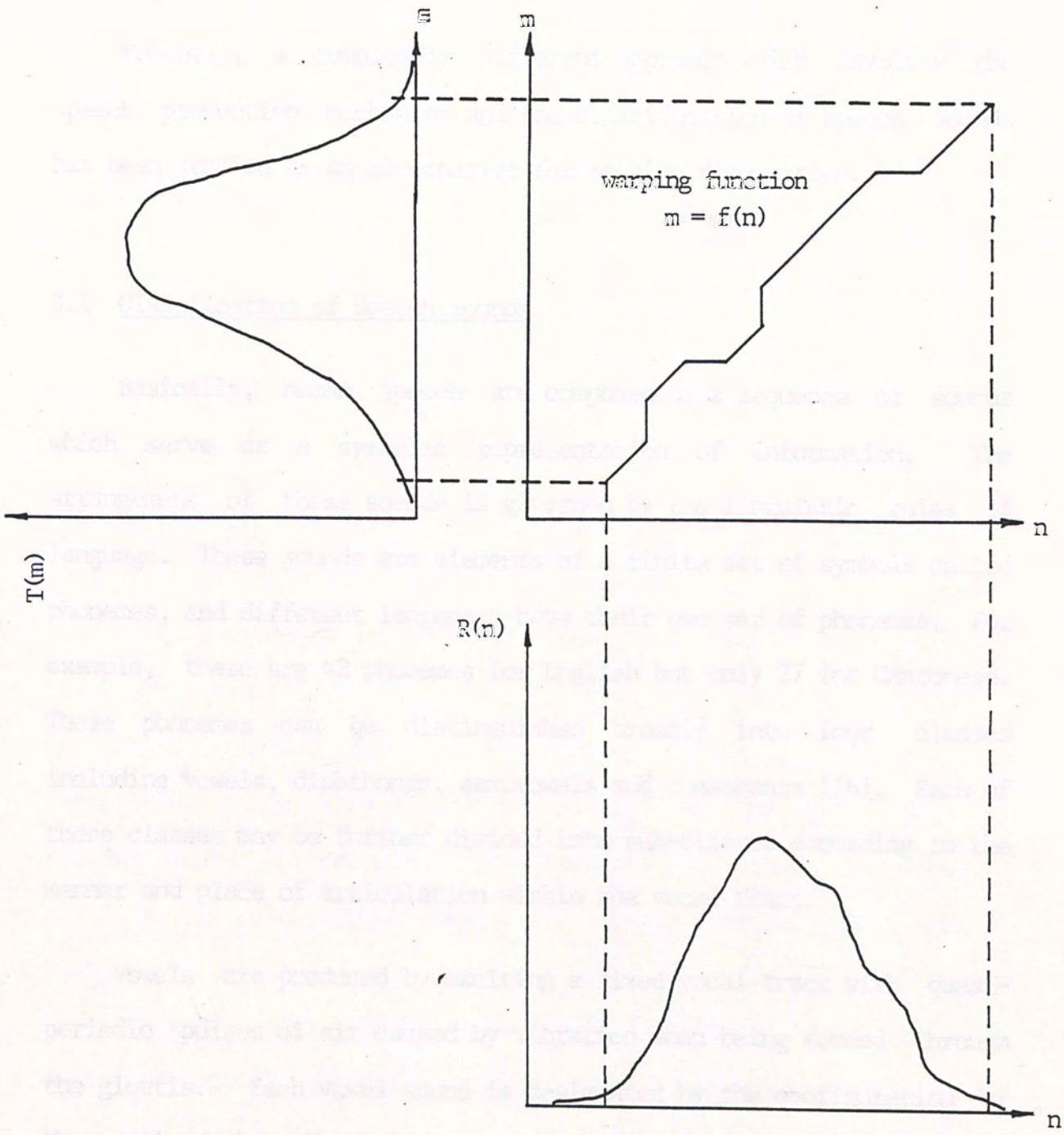


Figure 2.1 A typical warping function

CHAPTER 3 SPEECH RECOGNITION BASED ON PHONETIC STRUCTURES

Recently, a completely different approach which involves the speech production mechanism and the classification of speech sounds has been studied as an alternative for machine recognition.

3.1 Classfication of Speech Sounds

Basically, human speech are composed of a sequence of sounds which serve as a symbolic representation of information. The arrangement of these sounds is governed by the linguistic rules of language. These sounds are elements of a finite set of symbols called phonemes, and different languages have their own set of phonemes. For example, there are 42 phonemes for English but only 27 for Cantonese. These phonemes can be distinguished broadly into four classes including vowels, diphthongs, semivowels and consonants [16]. Each of these classes may be further divided into sub-classes according to the manner and place of articulation within the vocal tract.

Vowels are produced by exciting a fixed vocal tract with quasi-periodic pulses of air caused by vibration when being forced through the glottis. Each vowel sound is designated by the configuration of the vocal tract. For instance, in pronouncing vowel /a/, the vocal tract is constricted at the back by the tongue and is open at the front with the mouth wide opened. Conversely, vowel /i/ is pronounced by raising the tongue towards the palate, thus causing a constriction at the front while increasing the opening at the back of the vocal tract. The class of vowels includes /a/, /e/, /i/, /ɔ/, /u/, etc.

Diphthongs are defined as the combinations of two vowels. They are produced by first configuring the vocal tract corresponding to one vowel and then varying the shape towards the other. The phonemes /in/, /ou/, /ai/, /ei/, ..., etc. are belonging to this category.

The sounds /w/, /l/ and /r/ are called semivowels because of their vowel-like nature. They are not classified as vowels since their vocal tract configurations vary during pronounciation. The acoustic characteristics of these sounds are strongly influenced by their neighbouring phonemes in the context.

The class of consonants consists of nasals, stops, fricatives and affricates. The nasal consonants /m/, /n/ and /ŋ/ are produced by glottal excitation with sound being radiated at the nostrils. They are distinguished by the place at which a constriction is made. For /m/, the constriction is at the lips; for /n/, it is at the back of the teeth; whereas for /ŋ/, it is just before the velum. The fricatives, on the other hand, like /f/, /s/ and /sh/, are produced by forming a constriction at some point of the vocal tract and forcing air through the constriction at a high frequency to produce turbulance, thus creating a broadband noise to excite the vocal tract. The stop consonants, which are also called plosives, are produced by making a complete closure somewhere in the vocal tract, building up pressure behind the closure and releasing the pressure suddenly. Since the stops are dynamical in nature, their properties are highly influenced by the vowel that follows. The affricates, unlike the others, are in fact the combinations of two consonants, and phonemes /ts/ and /dz/ are some of the members in this class.

3.2 Phonetically-Based Recognition Systems

A speaker-independent digit-recognition system based on the acoustic features of the utterances was first built by Sambur and Rabiner in early 1975 [7]. The fundamental of the recognition algorithm was to represent the ten English digits as a sequence of phonetic labels as listed in Table 3.1. A set of robust measurements was used to classify the phonemes of an input utterance into one of the six specifically assigned categories. The measurements that had been exploited include zero-crossing rate, energy, normalized error and pole frequencies, with the latter two being obtained from a two-pole model linear predictive coding analysis. Each of these parameters can be used, individually, to classify a speech sound into one of the six classes. Therefore, when the classification results of all the measurements are 'intelligently' combined together, an enhanced classification performance can be achieved. A decision algorithm in the form of a tree structure was specifically designed for speaker-independent recognition of the ten digits with phonetic labelling. The parameters were checked when the decision flow traced down the most probable branch so as to arrive at the decided digit. This system was evaluated by carrying out two tests. An average error rate of 2.7 - 5.6 percent was obtained.

Despite the fact that high recognition rate can be achieved, a set of robust measurements is required which made the system rather complicated. Although the counting of zero-crossings is simple, the calculation of normalized error and pole frequencies using LPC analysis demands great computational complexity and is time-consuming. The recognition speed was not mentioned, but it is believed that real-

time processing would be very difficult unless special-purpose hardware was employed. Another disadvantage of this system is that since the decision algorithm is specially designed for the recognition of the ten digits only, an expansion of the vocabulary will inevitably involve a modification of the whole decision tree structure. Neither of these are desirable for practical implementation of a recognition system with daily applications.

Nearly ten years later, Vysotsky developed another speaker-independent discrete utterance recognition system that also made use of the phonetic structure of the word [8]. The three principal components of this system are: (i) using a deterministic strategy to give a crude phonetic structure for each utterance; (ii) using a set of parameters to statistically classify each of the segments; and (iii) carrying out a matching procedure between the utterance parameter set and that of the reference words having the same structure. The speech samples were divided into frames of 12.8ms with 50 percent overlapping after end-pointing. In each frame, nine measurements were made based on the energy contour, the zero-crossing rate associated with the noise-like component of consonants, and the proportionality of the first and second formant frequencies of the vowels. According to these nine measurements, each frame was labelled as one of the four types of segments, namely unvoiced pause, fricative-like segment, transition segment, and vowel-like segment. These frame labels were then transformed into a sequence of labelled segments representing the pattern of a word by using a set of rules for determining the minimum acceptable duration (in frames) of each type of segment. Every word might be represented by many patterns due

to the variabilities in pronunciation of different speakers. In addition, a second stage of processing was carried out in which each labelled segment was characterized by a more detailed parameterization. The parameters included the energy distribution in the segment, the slopes, the mean and the maxima of energy and zero-crossing rate, as well as the normalized duration of the segment.

In the reference set, the parameter vectors of all tokens with the same segment sequence were stored in the same table. Thus, the reference prototypes of different words might be stored together corresponding to a common pattern. Conversely, a given word might have several different prototypes in different tables corresponding to different patterns. The reference set of each word consisted of the mean and variance of the parameter vectors which made up the training set of tokens of the word. In the training mode, if the pattern existed and the reference set of the word was present in the table, it would be updated by the input parameters. If the pattern existed, but there was no reference prototype for that particular word, the new parameter set would be appended to the table. However, if the pattern did not exist at all, a table would be created.

When testing with an unknown input, the feature vector of the test word was compared with all the references in the table acquiring the same pattern as this utterance. A distance measure was used for comparison in which a weight coefficient was added to emphasize the vowel segment and the segments preceding the vowel because these segments were much more informative. The utterance was rejected if the distance measurements did not satisfy the pre-defined conditions

based on some chosen thresholds. It was also rejected if the pattern did not exist in the catalog of the table names. In addition, comparisons were not performed for those reference sets which come from only one occurrence of the word during training.

The system was evaluated on a 20-word vocabulary consisting the ten digits (0 - 9) and ten control words. There were totally 117 patterns and associated tables for the whole vocabulary, but the ten digits were found in only 22 of these tables. It was discovered that polysyllabic words has a much wider distribution across the patterns than the monosyllabic words. An accuracy score of over 95 percent was obtained. When considering the digits only, around the same recognition score was obtained but with a slightly higher error rate and lower rejection rate.

The recognition results of Vysotsky's system are quite satisfactory, but the preprocessing and parameterization of the speech signal involved are rather complicated. The requirement of large number of parameters to represent each segment demands large storage and long processing time. Furthermore, the system requires a relatively large set of training data due to the fact that several patterns are needed to represent the same word. Modifications had been made in the parameterization and the training scheme [9] for processing telephone quality speech and effecting reduction in memory requirements. Although the memory size was significantly reduced, the complexity of feature extraction still maintains because seven frequency bands were employed for energy and zero-crossing measurements and there was roughly 1 percent drop in performance.

More recently, a network-based isolated digit recognizer was developed within a general framework for phonetically based speech recogniton [10]. This system comprises three major part: (i) a finite-state pronunciation network; (ii) a set of acoustic pattern matchers; and (iii) a dynamic programming search algorithm. A pronunciation network is a collection of nodes interconnected by labelled and directed branches. Branches represent classes of acoustic-phonetic segments and a path through the network describes the pronunciation of a vocabulary item as a sequence of such segments. Table 3.2 shows the primitive segment classes used to model the eleven digits ("one" - "nine", "oh", and "zero"). The basic digit models employ 37 branches with 25 distinct acoustic-phonetic labels. The boundaries of the segments are defined in terms of three types of prominent acoustic events and a small number of duration-based segmentation rules. The first type is an abrupt change in the manner of articulation, such as occurs between the strong fricative 'S' and the vowel 'EH' in seven. The second type is the boundary between speech and silence whilst the third refers to the point where the third formant of the /r/ in zero reaches a local minimum.

Each of the primitive branches in a pronunciation network is connected with an acoustic pattern matcher. The segment matchers used in this recognition system were based on vector quantization (VQ) [23] of linear prediction spectra. Each segment class was represented as a sequence of 1 to 3 VQ codebooks. The codebooks for a matcher were created by clustering LPC spectra from labelled training tokens of the corresponding segment class. During recognition, a path was searched through the pronunciation network and the utterance was partitioned

into a corresponding sequence of segments. The input token was recognized as a specific word for which the best path was a possible pronunciation. Computing the best path usually proceeded in two steps. First, a dynamic programming algorithm was used to recursively compute, for each node in the pronunciation network and for each analysis frame of the input utterance, the score of the best partial path which terminated on that node at that time. The score was in terms of the distance measures of the LPC coefficients. Secondly, the node scores which act as "backpointers" were used to trace the best path backwards from the final node to the start node. The performance of the system was good and an average recognition rate of 97 percent was obtained. However, a major drawback of this approach is that it is time-consuming, both in computing matcher codebook distortions and path searching which involves dynamic programming. Typically, the distortion computation requires 32s and the path searching spends 19s. Another disadvantage, similar to the system built by Sambur and Rabiner, is that the vocabulary of the system cannot be expanded without system modifications.

In general, speech recognition system based on phonetic structures do not require temporal alignment. Therefore, dynamic time warping is not necessary and hence, less computation is needed as compared with those systems described in Chapter 2. However, the decisions in phonetic labelling usually depend on more than one parameter of the speech, and as a result, more efforts have to be put on parameterization which in turn increase the computational load. In addition, these recognizers normally possess complex system models due to the sophisticated decision strategies and these recognizers are not

suitable for monosyllabic languages since their phonetic structures are too simple to provide enough features for recognition.

After reviewing many of the existing recognition systems, it seems that none of them are simple enough for real-time implementation without using special-purpose and expensive hardware. The primary goal of this project is to design an efficient recognition algorithm for Cantonese which is basically a monosyllabic language. Therefore, systems based on phonetic structures will not be suitable. Whereas the other systems using spectral features, LPC and cepstrum, normally require too many calculations in feature extraction. Thus, the filter bank approach seems to be most appropriate if time warping process can be eliminated. In Chapter 4, we shall describe a novel recognition scheme for monosyllabic languages which uses the energy-time profiles of a word at different frequency bands as the parameter for recognition.

Digit	Sequence of Sound Classes
0	VNLC -- FV -- VLC -- BV
1	VLC -- MV -- VLC
2	UVNLC -- FV -- BV
3	UVNLC -- VLC -- FV
4	UVNLC -- BV -- MV
5	UVNLC -- MV -- FV -- VNLC
6	UVNLC -- FV -- UVNLC
7	UVNLC -- FV -- VNLC -- FV -- VLC
8	FV -- UVNLC
9	VLC -- MV -- FV -- VLC

VNLC - voiced, noise-like consonant
UVNLC - Unvoiced, noise-like consonant
VLC - Vowel-like consonant
FV - Front vowel
MV - Middle vowel
BV - Back vowel

Table 3.1 Sound Classes Characteristic of the digits [7]

Digit	# Segments	Segment Labels
Oh	1	OW
1	3	W AH N
2	2	TR UW
3	3	TH R_i IY
4	3	F AO R_f
5	4	F AY_1 AY_2 V
6	4	S IH KS S
7	5	S EH V AX N
8	3	EY TS TR
9	4	N AY_1 AY_2 N
Zero	5	Z IY R_{m1} R_{m2} OW

Table 3.2 Acoustic-Phonetic Segment Classes Used in Pronunciation Networks [10]

CHAPTER 4 SPEECH RECOGNITION USING ENERGY-TIME PROFILE

4.1 Phonetic Characteristics of Cantonese

Literary language of Chinese is basically written in hieroglyph with each character (or word) being the smallest unit built up by radicals and is uniquely adopted. For spoken language, there are, however, many dialects in China, but they are all monosyllabic tonal languages. The word 'tonal' means that the same syllable may represent different characters when it is pronounced in different tones. Cantonese is one of the most commonly used dialect in Southern China and in Hong Kong with the population of speakers in the order of tens of millions. In this project, we have investigated a novel recognition scheme which is particularly suitable for monosyllabic languages. This recognition system has been evaluated extensively using Cantonese as the test language.

Basically, Cantonese contains 27 phonemes including ten vowels, three nasals, and three stops. In Cantonese, each syllable of a particular word is regarded as made up of an initial (聲母) and a final (韻母) which may be expressed by the following "phonetic equation",

$$\text{syllable} = \underbrace{\text{consonant}}_{\text{initial}} + \underbrace{\text{vowel} + \text{consonant}}_{\text{final}}$$

The initials actually correspond to leading consonants and the consonants appear in the initial and final are optional. There are altogether 19 initials for Cantonese containing plosives /p, t, k, b,

d, g/, nasals /m, n, ŋ/, semivowels /l, j, w/, fricatives /f, s, h/, affricates /ts, dz/ and the so-called plosive-semivowels /kw, gw/. But there are exceptions in which the syllables do not begin with consonant, for example, /a$_1$/[1] (鴉), /ou$_3$/ (奧), and /ŋ$_5$/ (伍). Finals are either single vowels, diphthongs, vowels plus nasals, or vowels plus stops. The nasals and stops are collectively called final consonants which include /m/, /n/, /ŋ/ and /p/, /t/ and /k/. Among the 53 finals, the two voiced-nasals /m̩, ŋ̩/ are different from others in that they do not contain a vowel. A summary of the finals is given in Table 4.1.

The make-up of a Chinese syllable is not only determined by its initial and final, but also by the musical pitch usually called the tone. In general, there are four basic classes of tones, namely the level (or even) tone (平聲), the rising (or ascending) tone (上聲), the going (or departing) tone (去聲), and the entering tone (入聲). These four tones can be further subdivided into the upper and lower series in terms of their pitch levels, hence giving a total of eight tones. Cantonese, however, has an extra middle-level entering tone, therefore comprising nine tones altogether. All of these tones may be represented by a musical scale as shown in Figure 4.1. It can be seen that the musical values of the upper, middle and lower entering tones are identical with the upper level, upper going and lower going tones respectively. The only difference is that the entering tones always end in stops /p/, /t/ or /k/ while others end in either vowels or nasals. It is worth to note that not every tone of a single syllable

[1]The subscribed number refers to tone.

will correspond to a word with verbal meaning. For example, the nine different tones of /si/ give nine meaningful words whereas for /din/, four of the nine tones do not exist.

si_1	詩	din_1	奠
si_2	史	din_2	典
si_3	試	din_3	墊
si_4	時	din_4	X
si_5	市	din_5	X
si_6	事	din_6	電
sik_7	色	dip_7	X
sik_8	錫	dip_8	X
sik_9	食	dip_9	碟

In order to perform speech recognition on Cantonese, it is essential to possess some fundamental knowledge of its phonetic characteristics. However, the details of the language such as contextual variation of tone (變調), formation of /tsi_4/ (詞 - string of characters bearing specific meaning), semantic structures are excluded because we intend to build a recognition system which is simple but efficient and can be implemented by low cost microcomputers. It will be illustrated in the later sections that, in our design, the detection algorithm, the classification procedure as well as the elimination of the time warping process all make use of the special phonetic features of the monosyllabic language.

4.2 System Configuration

A speech recognition system using the energy-time profiles (ETP) of a word at different frequency bands as the parameter has been developed. The feature vectors of this system is similar to the filter bank approach, except that they are all equal in dimension rather than having various sizes. This eliminates the warping process usually required for temporal alignment and hence a much simpler recognition scheme is possible. In addition to its low cost, one good reason for using filter bank analysis is because the ear seems to process speech sounds using a structure similar to filter bank.

Figure 4.2 is a block diagram showing the overall structure of the recognition system. On receipt of an unknown utterance, the beginning and end of the token is first determined by the endpoint detection algorithm. After that, the word is classified as a member of either the fricative-initial (FI) or the voiced-initial (VI) group depending on its acoustic features at the beginning region. The speech signal is then passed through a series of bandpass filters and a matrix of energy-time profiles is created. This matrix, serves as the parameter vectors of the input utterance, is compared with the stored references by calculating the distances among them. The template whose distance is the minimum within the whole reference set will be taken as the uttered word provided that this minimum distance exceeds the next minimum distance by a predefined threshold. Otherwise, the input will be rejected.

Details of each functional block shown in Figure 4.2 will be described in the following sub-sections.

4.2.1 Endpoint Detection

The performance of a speech recognition system critically depends on the accuracy of locating the beginning and ending of the input utterances. Besides, the computation for processing the speech signal will also be reduced if the endpoints are precisely located. Therefore, it is essential to determine as accurately as possible the endpoints of an utterance such that all significant acoustic events within the word are included. The endpoint detection techniques can be broadly classified into three main approaches, namely explicit, implicit, or hybrid [24]. The explicit technique locates the endpoints prior to and independent of the recognition and decision stages of the system. Whereas for the implicit method, the endpoints are determined during the recognition and decision process. The hybrid technique, as its name suggests, incorporates aspects from both explicit and implicit methods. In this system, we have employed a simple explicit endpoint detector that merely depends on segmental 'energy' and zero-crossing rate [25].

Figure 4.3 shows a flow chart of the endpoint detection algorithm. The speech waveform is first bandpass filtered at 100Hz - 3.3kHz to simulate the telephone line environment prior to sampling at 8kHz. The sampling buffer is 1.5s which means that the utterance should be finished within this period, and in fact, this is quite sufficient for any single isolated word. The entire speech sequence is divided into 10ms segments giving altogether 150 blocks and each block contains 80 samples. Let n denotes the segment number, and $s_n(i)$ represents the i^{th} sample in the n^{th} segment. The segment 'energy' is defined as,

$$E(n) = \sum_{i=1}^{80} \left| s_n(i) \right| , \qquad n = 1, 2, \ldots, 150 \tag{4.1}$$

which is the sum of the absolute magnitudes of the speech samples in the n^{th} segment. The absolute magnitudes are used instead of the actual energy in order to minimize computations. The maximum block energy is then given by,

$$E_{MAX} = \text{Max}(E(n)) \qquad \forall n \tag{4.2}$$

where Max(.) gives the maximum value in the list of argument (.).

The zero-crossing rate, also on segment basis, is defined as the total number of zero crossings occurred in each block and can be expressed as follows,

$$Z(n) = \sum_{i=1}^{80} 1/2 \left| \text{Sgn}[s_n(i)] - \text{Sgn}[s_n(i-m)] \right| , \qquad n = 1, 2, \ldots, 150 \tag{4.3}$$

where $\text{Sgn}[s(i)] = \begin{cases} 1 & \text{if } s(i) > 0 \\ 0 & \text{if } s(i) = 0 \\ -1 & \text{if } s(i) < 0 \end{cases}$

and $s_n(i-m)$, $m \geq 1$, refers to a non-zero sample that is m samples before the present one.

The presence of speech signal can usually be detected if the energy is above a certain level and if the zero-crossing rate is significantly higher than that occurred during silence. Of course, the accuracy of the detection relies very much upon the algorithm and the chosen thresholds. In order to take into account of the effect of

speaker- and time-dependence as well as the variation of background noise, a self-normalization technique is utilized in which the energy and zero-crossing thresholds employed are obtained from measurements of the utterance during test rather than using fixed values. First of all, we have assumed that during the first 100ms of all the input utterances, there is no speech present. Thus, the statistics of the background silence can be computed from these ten segments. The energy during silence (channel noise only) therefore equals to

$$E_{SIL} = \frac{1}{10} \sum_{n=1}^{10} E(n) \tag{4.4}$$

Three energy thresholds are then defined as the function of the maximum block energy E_{MAX} and the silence energy E_{SIL} and are given by,

$$E_{UT} = 0.2 \times (E_{MAX} - E_{SIL}) + E_{SIL} \tag{4.5a}$$

$$E_{MT} = 0.1 \times (E_{MAX} - E_{SIL}) + E_{SIL} \tag{4.5b}$$

$$E_{LT} = 0.01 \times (E_{MAX} - E_{SIL}) + E_{SIL} \tag{4.5c}$$

The zero-crossing threshold, on the other hand, is chosen as the minimum of a fixed value (25 crossings per 10ms) and the sum of the mean plus twice the standard deviation of the zero-crossing rate obtained during silence, i.e.,

$$ZC_T = \text{Min}(25, \overline{ZC} + 2 \times \sigma_{zc}) \tag{4.6}$$

where $\overline{ZC} = \frac{1}{10} \sum_{n=1}^{10} Z(n)$

and $\sigma_{zc} = \left\{ \frac{1}{10} \sum_{n=1}^{10} (Z(n) - \overline{ZC})^2 \right\}^{\frac{1}{2}}$

The reason for choosing the zero-crossing threshold in this way is that speech is fairly certain to be present in a segment when its zero-crossing rate is sufficiently larger than that occurred in the silence region. However, if the value of the threshold given by (4.6) is too high due to some additional noise, we may mislocate the beginning segment. Thence a fixed value, 25, is introduced to limit the upper bound of the zero-crossing rate threshold. The typical values of the energy and zero-crossing rate thresholds of the Cantonese digit "1" are depicted in Figure 4.4.

Having calculated all the thresholds, the detection for the beginning and end of the speech can be carried out. We first locate the segment having the maximum 'energy' and then search backwards until the segment 'energy' is smaller than the mid-threshold, E_{MT}. Let this block be j. During this process, the beginning of the vowel region is also determined and the block whose 'energy' is just above the upper threshold E_{UT} is denoted by 'VOWB'. The following four conditions are being checked in sequence in order to locate the beginning block of the utterance :

(1) $E(j) < E_{LT}$ and $E(j-1) < E_{LT}$

(2) $E(j) < E_{LT}$ and $Z(j) < ZC_T$

(3) $(E(j) - E(j+1)) > E_{LT}$ and $Z(j) < ZC_T$

(4) $(VOWB - j) > 10$

The first two conditions states that if two consecutive segments have 'energy' smaller than the lower threshold E_{LT}, or if both the 'energy' and zero-crossing count are lower than their respective thresholds, then we can conclude that no speech is present in that region. The

third condition is applied to exclude the mouth noises at the beginning of the utterance while the last one limits the length of the initial region to a maximum of ten blocks. By observing the E(n) and Z(n) of some tokens with very long initial regions, we find that it is not necessary to take into account the entire initial period since all the useful information are contained in a much shorter interval preceding the vowel. Therefore, we have restricted the number of blocks of the initial to ten in order to cut down the processing time of words having very long initials while at the same time preserve all the acoustic characteristics of the initial region. If either one of these four conditions is satisfied, the block (j+1) is marked as BEGIN indicating the beginning of the input utterance. On determining the end of the word, we also start from the block of maximum 'energy' and search forwards until the segment 'energy' is lower than E_{LT}. The block immediately in front of this is marked as END denoting the end segment of the word. No other criterion is employed because by searching forward from the middle of the word has automatically eliminated the breath noises at the end of the utterance that might cause uncertainties to the ending. The simplicity of this endpointing algorithm is a result of the properties of monosyllabic words, whose phonetic structure is always in the form of "initial consonant + vowel + final consonant". Hence the middle section of a word will always have high energy which provides a good starting point for searching the beginning and end of the word. The algorithm has been found to be very successful and is inherently capable of performing correctly in any reasonable acoustic environment.

4.2.2 Classification

The primary goal of introducing the classification scheme is to reduce the number of comparisons during template matching. If an input token is classified into a particular group, it will only be compared with the reference templates within that group, whilst the references in the other groups are automatically screened out without any actual comparison and hence less computation is required.

As mentioned previously, Cantonese is a monosyllabic language whose phonetic structure is of the form 'Initial + Final'. For the sake of simplicity, we try to classify the vocabulary into small subgroups according to the phonetic labelling of their initial regions only. Unfortunately, there are still altogether five different types of initials for Cantonese including semivowels, plosives, fricatives, affricates, and the so-called plosive-semivowels. To distinguish all of them by analytical methods is by no means a trivial task and often involves very complicated algorithms. However, it is noted that the plosives, semivowels, and the plosive-semivowels all have a common feature in that they possess low frequency components whereas for the fricatives and affricates, they are noise-like and are of much higher frequencies. For those syllables that do not contain an initial consonant, their properties are very similar to the semivowel initials. This observation leads to a simple classification scheme which uses the zero-crossing rates of the utterances. The entire vocabulary is, therefore, classified into two groups, one being the voiced-initial (VI) group whose members possess low zero-crossing rates; and the other being the fricative-initial (FI) group whose members all have very high initial zero-crossing rates. Additionally,

since most of the members within the VI group are characterized by a sharp rise in energy, an utterance is also classified to the VI group if this condition is detected. Table 4.2 shows the phonetic transcriptions and the corresponding initial labelling of the ten Cantonese digits. The FI group contains the digits 3, 4, 7, 10 whilst the rest fall into the VI group.

The classification procedures are summarized in the flow chart as shown in Figure 4.5. The thresholds LEN_T, ZCR_{LT} and ZCR_{UT} are all chosen experimentally and have significant effects on the accuracy of the classification. Unlike the endpoint detection algorithm, all thresholds used in the classification procedure are unchanged assuming that the acoustic properties of the phonemes are independent of the speakers. The typical values used in our tests for LEN_T, ZCR_{LT} and ZCR_{UT} are 3, 25, and 35 respectively. All the measurements and decisions are made within the initial region. If the length of this beginning portion is very short, which means that there is a rapid change in energy, then it must have a voice-like initial. Otherwise, the zero-crossing rates are checked to determine whether it belongs to the VI or FI group. For an input token, if five or more consecutive segments in the initial region have Z(n) larger than the upper threshold ZCR_{UT}, or if 75 percent or higher of the segments in the initial region have Z(n) large than ZCR_{UT}, then it must belong to the FI group. Similar conditions are also applied for the decisions of VI words, but with Z(n) being compared to the lower threshold ZCR_{LT}. If all the conditions are not satisfied, the input utterance is said to be unclassified and it will be compared with the whole vocabulary during template matching.

This classification scheme has found to be very efficient throughout our tests. Indeed, less than ten percent of all the tokens that have been studied were unclassified and none of them was identified to the wrong group.

4.2.3 Feature Extraction

Currently, the parameters that have been used to represent the acoustic features of a speech signal include zero-crossing rate, energy, spectral composition and the linear prediction coefficients. Most recognition systems extract these parameters on a fixed duration basis, that is, the input signal is divided into a number of equal-length time frames. This implies that the size of the feature vectors will vary for different utterances even for the same word, and hence, warping technique must be employed. In our method, we choose to use the energy-time profile as the parameter for recognition. Each uttered word is represented by several vectors of fixed number of parameters so that dynamic warping is unnecessary during matching.

The input speech samples are first sent to a bank of five bandpass filters with passbands at (i) 150-500Hz, (ii) 500-850Hz, (iii) 850-1.2kHz, (iv) 1.2k-1.8kHz, and (v) 1.8k-3.2kHz. The number of filters employed has a crucial effect on the overall performance of the system. On one hand, we want to have as many filters as possible such that all spectral features especially formant frequencies can be captured. However, the more filters we use, the larger the amount of computations needed. Hence, there is a trade-off between these two

factors and eventually we have chosen to use five bandpass filters which seems to give a good compromise between them. These filter bands are carefully selected to have uniform spacing and finer division in the most important range of 150-1.2kHz. It has been shown that a uniform frequency spacing in the low frequency region is desirable for filter bank recognition systems [18]. The design specifications for these filters are given in Table 4.3, and the plots of the filter characteristics are shown in Figure 4.6. The outputs from these filters together with the wide-band signal form six different sequences of the utterance. Each sequence is then evenly divided into a fixed number (N) of segments, with 50 percent overlapping, as depicted in Figure 4.7. Hence for different words, the segment length will vary according to the utterance's duration. Let $E_q(k)$ be the energy of each segment k=1, 2, ..., N, in the sequence q=1, 2, ..., 6, which is given by the sum of the squares of the magnitude of each sample within that frame, i.e.,

$$E_q(k) = \sum_{i}^{SL} (s(i))^2 \qquad (4.7)$$

where SL = segment length. Let E_{1M} be the maximum energy of the wide-band sequence, then the self-normalized energy of each segment

$$\overline{E_q}(k) = \frac{E_q(k)}{E_{1M}} \qquad (4.8)$$

is calculated which forms the energy-time profiles (ETP) of the word. Therefore, each token is now characterized by six vectors of ETP, each with N elements. Each of these vectors actually represents the

temporal variaton, on a segment basis, of the amplitude of the spectral envelope for that particular frequency band. We have chosen N=16 in all our tests on an ad hoc basis. The vectors of ETP so created for the input utterance is used to compare with the reference templates, to find the correct word by minimum distance measure.

It can be easily seen that when dividing the input signal into a fixed number of segments for extraction of feature parameters in the time domain, a linear compression or extrapolation on the time axis is actually performed. Obviously, this is not as good as the nonlinear time warping process because the ways of speaking might not vary linearly. However, in general, the speaking rate variation of monosyllabic words is much less than that of the polysyllabic words. Besides, the energy profiles of the utterances of a particular monosyllabic word maintains a similar shape even when their durations are different. This phonetic characteristic enables us to apply such a simple and direct method for temporal alignment so that the time-consuming dynamic programming for time warping can be eliminated. In addition, the computations required to calculate the ETP matrices for recognition is very much less than that of using linear prediction coefficients.

4.3 Distance Measure

The similarity between the reference pattern and an input pattern can be defined as a function of the vector distance between the feature vectors representing the two patterns, and is usually called the distance measure. Therefore, the smaller the distance measure,

the greater the similarity. It is essential that the distance measure D(x,y) between two frames of speech parameters x and y should always be positive and symmetric [17], i.e., $D(x,y) > 0$ for $x \neq y$ and $D(x,x) = 0$, and $D(x,y) = D(y,x)$. Furthermore, it should also be possible for efficient evaluation.

In our system, the distance D(w) between the ETP matrix of the test token and that of the template of the word w in the vocabulary is calculated by the formula,

$$D(w)[Sqr] = \sum_{q=1}^{6} \sum_{k=1}^{16} \frac{[\overline{E_q}(k) - R_w\overline{E_q}(k)]^2}{\overline{E_q}(k) + R_w\overline{E_q}(k)} \qquad (4.9)$$

where $R_w\overline{E_q}(k)$ is the normalized segment energy of the reference ETP matrix of the word w. This measurement can be considered as a modified Euclidean distance. The squared difference of each pair of segment energy is divided by their sum so that the difference is normalized nonlinearly. It is not surprising that the value of $\overline{E_q}(k)$ and $R_w\overline{E_q}(k)$ can vary in magnitude by several order with the low energy portion being at the beginning and end of a word and the high energy portion being contributed by the vowel in the middle. Obviously, if no normalization is adopted, the distance D(w) will be dominated solely by the large $\overline{E_q}(k)$ at the voiced region which is surely not preferable. This undesirable effect is overcome by using the normalization technique as described in (4.9). The difference is effectively normalized by the mean of $\overline{E_q}(k)$ and $R_w\overline{E_q}(k)$ so that the significance of both elements are taken into account.

In order to illustrate this normalization procedure more clearly,

let's look at an example. Consider two cases, (i) $\overline{E_q}(k)_1 = 0.2$, $R_w\overline{E_q}(k)_1 = 0.21$, and (ii) $\overline{E_q}(k)_2 = 0.2 \times 10^{-4}$, $R_w\overline{E_q}(k)_2 = 0.21 \times 10^{-4}$. The percentage of deviation in both cases is 5 percent. The normalized energies given in (i) and (ii) are typical values being obtained during the vowel period and at the initial region respectively. For case (i),

$$D(w)_1 = 0.01^2 \qquad \text{(no normalization)}$$

$$D(w)_{1N} = \frac{0.01^2}{0.41} \qquad \text{(with normalization)}$$

and for case (ii),

$$D(w)_2 = 0.01^2 \times 10^{-8} \qquad \text{(no normalization)}$$

$$D(w)_{2N} = \frac{0.01^2}{0.41} \times 10^{-4} \qquad \text{(with normalization)}$$

We can see that if no normalization is adopted, the ratio of distances $D(w)_1$ and $D(w)_2$ is simply the square of the ratio of the energy, i.e.,

$$\frac{D(w)_2}{D(w)_1} = \left\{ \frac{\overline{E_q}(k)_2}{\overline{E_q}(k)_1} \right\}^2$$

That means the difference of distance is magnified. As a result, the distance obtained from low energy portion will practically have no effect in the overall distance measure. However, with normalization, the ratio of the two distances is given by the ratio of the energy, i.e.,

$$\frac{D(w)_{2N}}{D(w)_{1N}} = \frac{\overline{E_q}(k)_2}{\overline{E_q}(k)_1}$$

Therefore, for the same percentage of deviation, the order of the distance measure follows the order of $\overline{E_q}(k)$. In other words, the larger the magnitude of the segment energy, the higher the degree of influence of the distance in the overall D(w). One may argue that the high energy portions are still more emphasized than low energy portions. Yet this is reasonable since the actual change in the high energy portions is much large than the change in low energy portions if the same percentage of deviation is considered.

The squaring in (4.9) may be omitted to save computations which results in an alternative way to compute D(w) by using

$$D(w)[Abs] = \sum_{q=1}^{6} \sum_{k=1}^{16} \frac{\left| \overline{E_q}(k) - R_w \overline{E_q}(k) \right|}{\overline{E_q}(k) + R_w \overline{E_q}(k)} \qquad (4.10)$$

In this circumstance,

$$D(w)[Abs] = \frac{0.01}{0.41} \qquad \text{(with normalization)}$$

for both cases (i) and (ii) which implies that the distance measure will be the same for both high or low energy if the percentage of deviation is the same.

The distance measure described in this section are employed during the procedure of template matching in order to determine the closeness of an input token to the templates in the reference set. The reference templates are created using various clustering techniques which will be discussed in detail in next chapter.

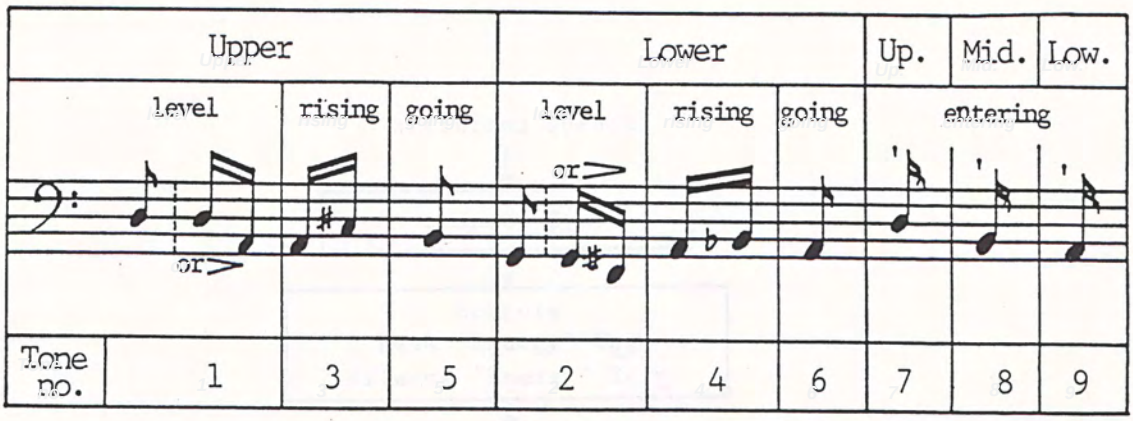


Figure 4.1 Representation of Cantonese Tones [26]

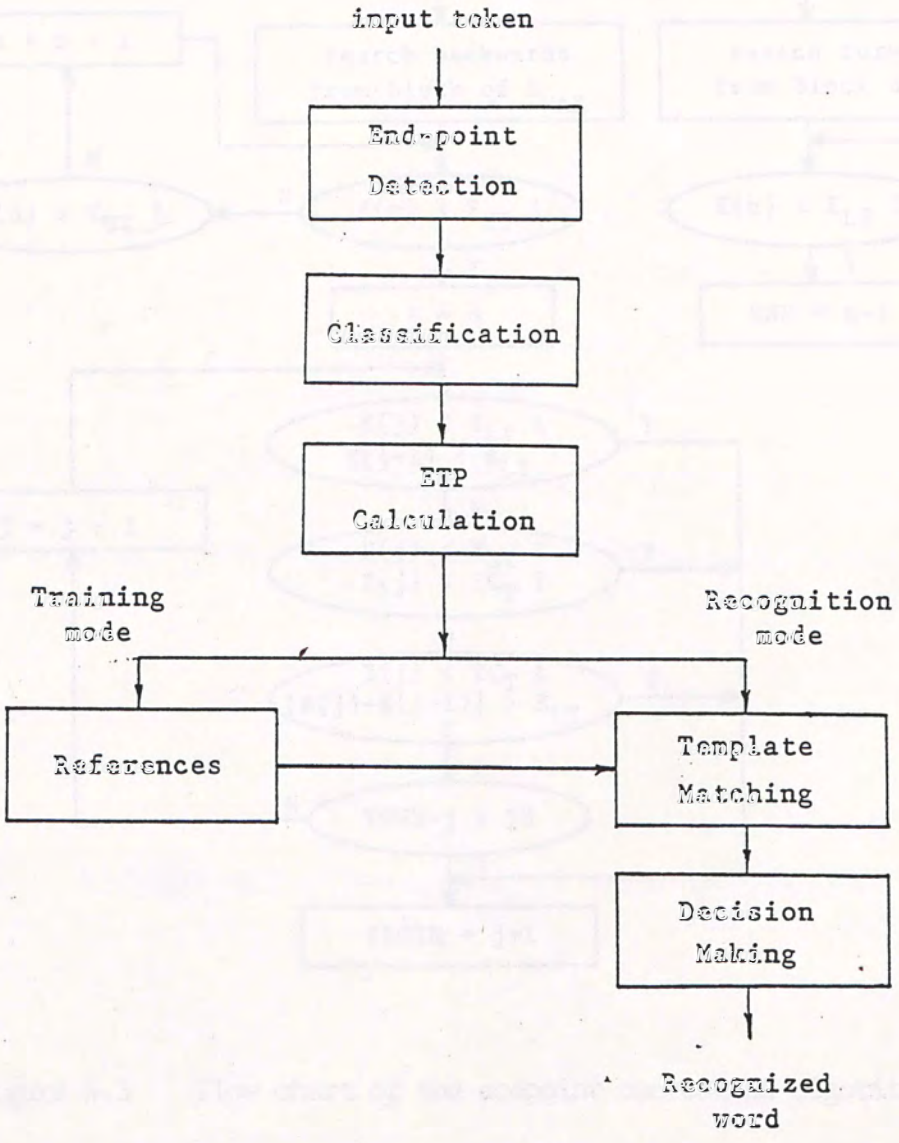


Figure 4.2 Overall structure of the recognition system

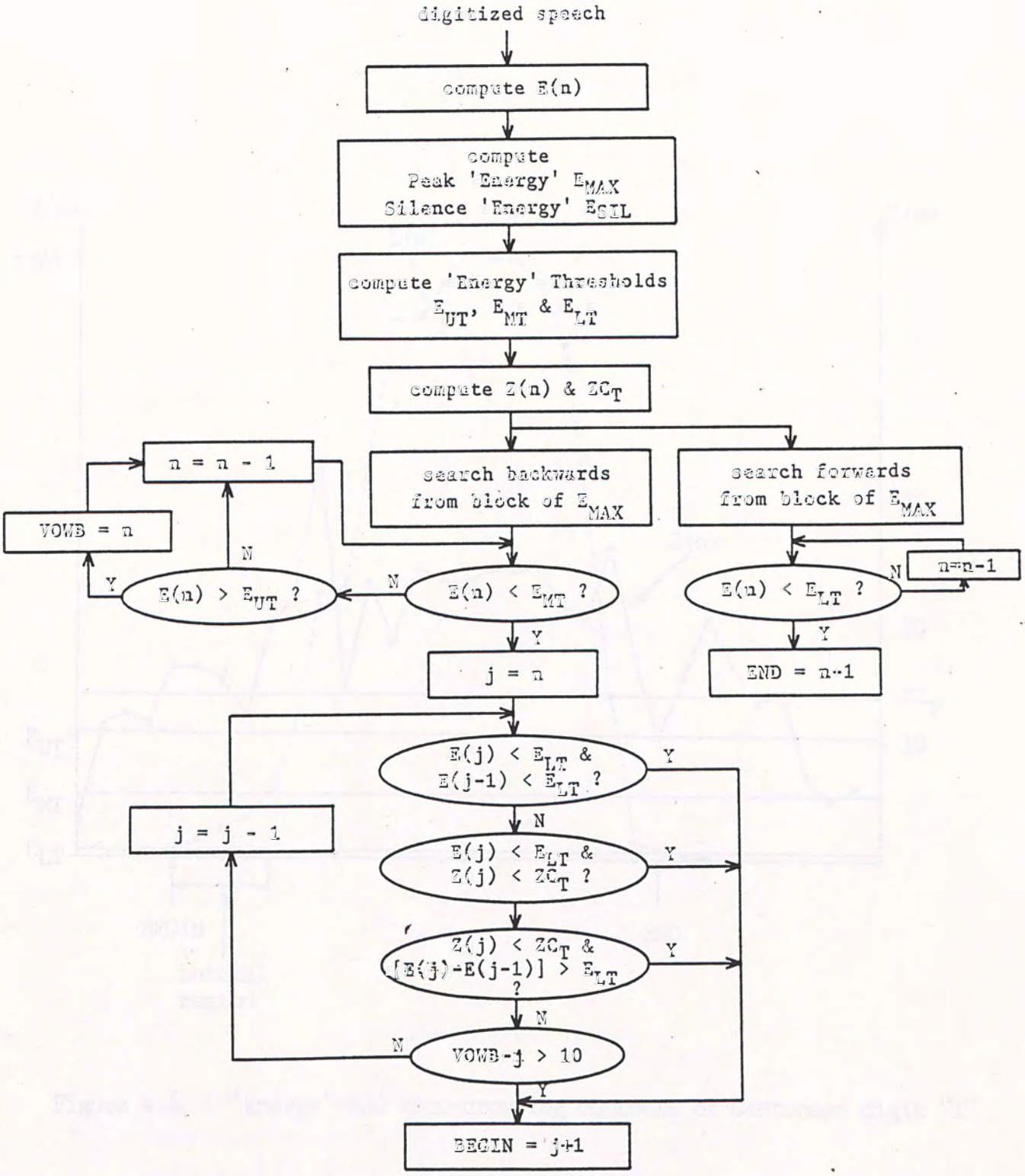


Figure 4.3 Flow chart of the endpoint dectection algorithm

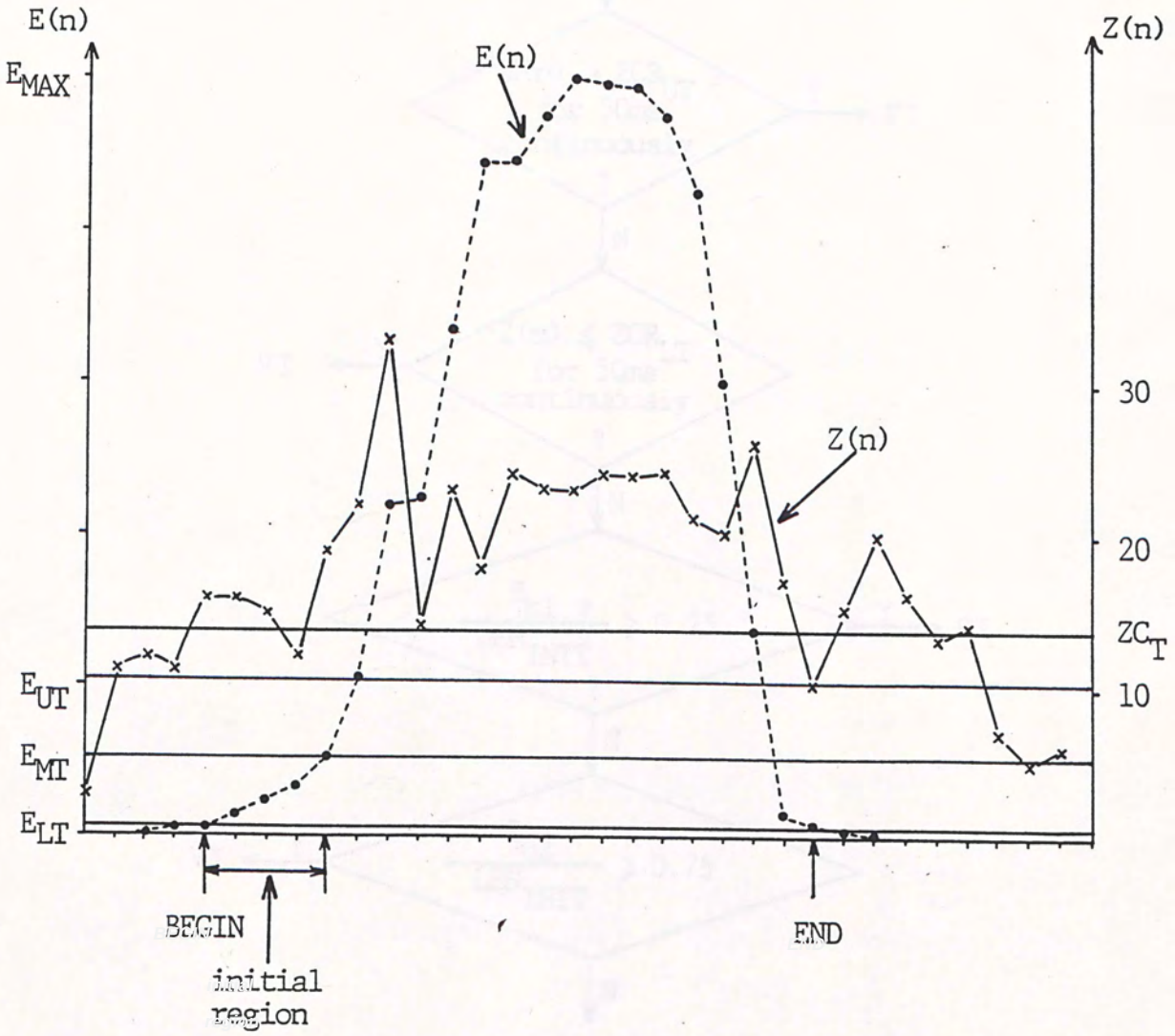
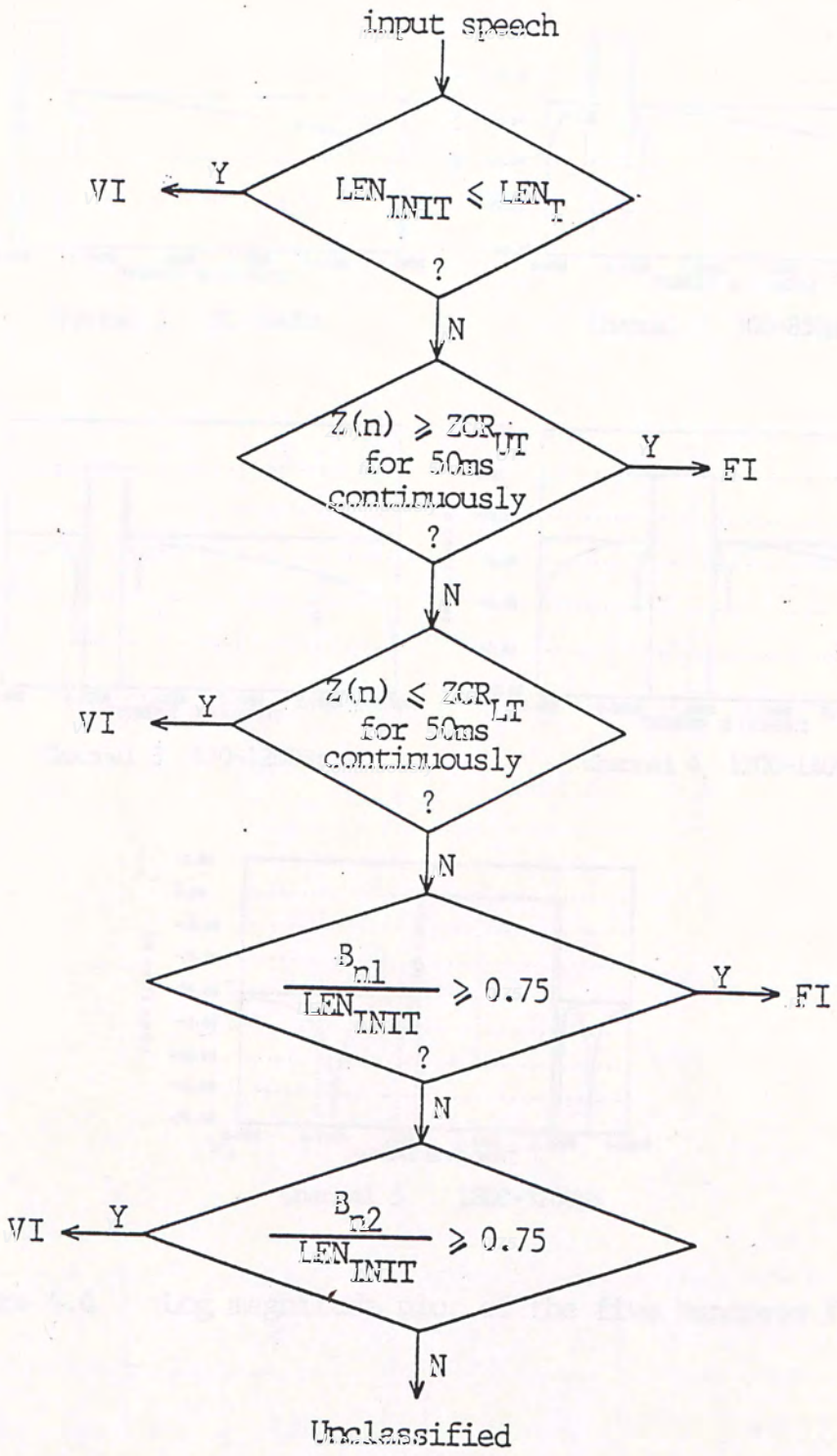
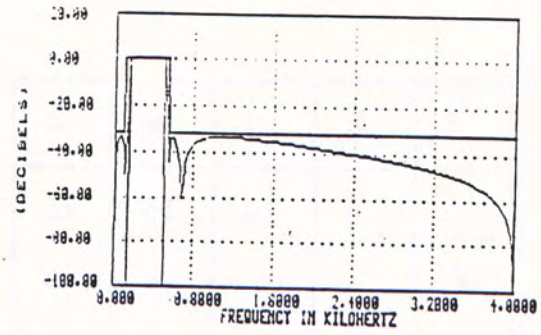


Figure 4.4 'Energy' and zero-crossing contours of Cantonese digit "1"

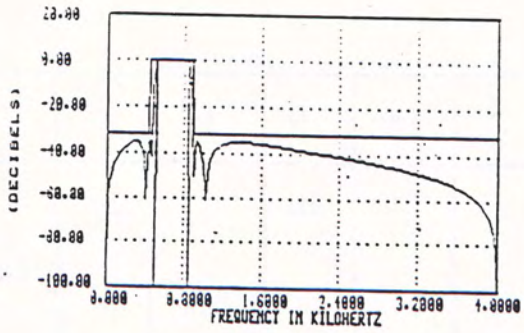


LEN_{INIT} = number of segments in initial region
B_{n1} = number of blocks in which $Z(n) \geq ZCR_{UT}$
B_{n2} = number of blocks in which $Z(n) \leq ZCR_{LT}$

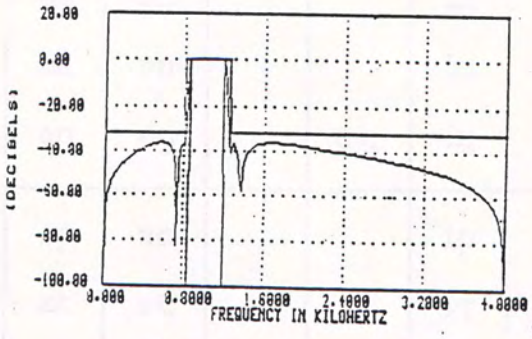
Figure 4.5 Flow chart of the classification scheme



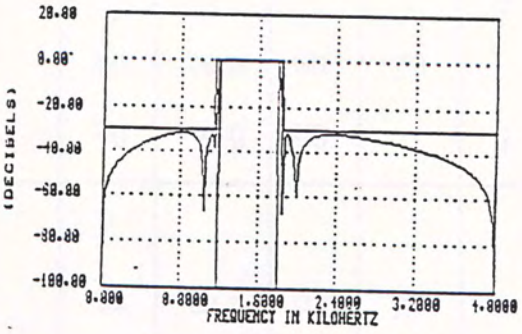
Channel 1 150-500Hz



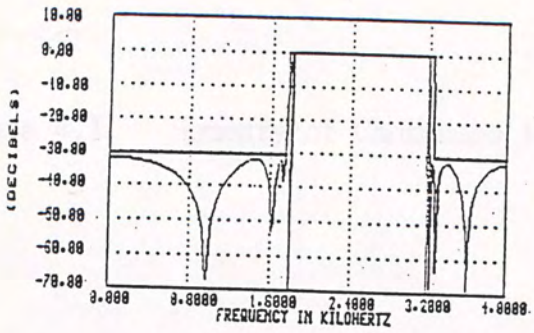
Channel 2 500-850Hz



Channel 3 850-1200Hz



Channel 4 1200-1800Hz



Channel 5 1800-3200Hz

Figure 4.6 Log magnitude plot of the five bandpass filters

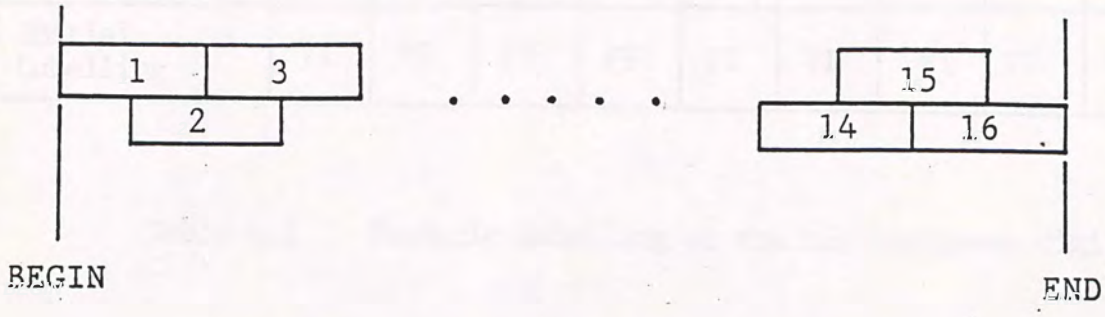


Figure 4.7 Segmentation of sequences

a	ɐ	e	ɛ	i	o	ɔ	œ	u	y	-	
ai au	ɐi ɐu	ei		 iu	 ou	ɔi	 œ y	ui			-i -u -y
am an aŋ	ɐm ɐn ɐŋ		 ɛ ŋ	im in iŋ		 ɔn ɔŋ	 œ n œ ŋ	 un uŋ	 yn	m̩ ŋ̩	-m -n -ŋ
ap at ak	ɐp ɐt ɐk		 ɛk	ip it ik		 ɔt ɔk	 œ t œ k	 ut uk	 yt		-p -t -k

Table 4.1 Summary of Cantonese Finals [26]

Digit	1	2	3	4	5	6	7	8	9	10
Phonetic Transcriptions	jɐt	ji	sam	sei	ŋ̩	luk	tsɐt	bat	gɐu	sɐp
Initial Labelling	VI	VI	FI	FI	VI	VI	FI	VI	VI	FI

Table 4.2 Phonetic labelling of the ten Cantonese digits

Channel No.	Lower Stopband Frequency (Hz)	Lower Passband Frequency (Hz)	Upper Passband Frequency (Hz)	Upper Stopband Frequency (Hz)	Stopband Ripple (dB)
1	100	150	500	550	-33
2	450	500	850	900	-35
3	800	850	1200	1250	-36
4	1150	1200	1800	1850	-32
5	1750	1800	3200	3250	-32

Table 4.3 Design specifications of the bandpass filters

CHAPTER 5 SYSTEM TRAINING AND EVALUATION

5.1 Clustering Techniques for Template Generation

A set of carefully chosen reference templates can enhance the performance of a speech recognition system, especially for speaker-independent environment. For speaker-trained recognizers, the importance of template generation is less significant. This is due to the fact that the variance between repetitions of the same word by the same speaker is relatively small. Hence only a few training tokens for each word, say, two or three, combined by a simple averaging process are enough to produce satisfactory references. On the contrary, for speaker-independent systems, the reference tokens usually come from a wide variety of talkers with different ways of saying words. In this case, simple averaging will provide a poor representation of the reference word. It has been shown that a single word should be represented by multiple templates for speaker-independent recognition in order to have high recognition accuracy. Extensive studies have been carried out to develop a generalized method for creating references [14, 27 - 29]. It was found that only those reference tokens which form a cluster need to be averaged to give a template. Among the various methods commonly used, the 'modified K-means' (MKM) clustering algorithm [14] is considered to be the most efficient one. It requires very few user defined parameters and the algorithm is basically independent of the number of reference tokens and the number of clusters to be created, which means that no modifications of the algorithm are necessary even if these parameters were changed. Because of these advantages, a clustering technique

similar to the MKM algorithm has been adopted in our system to generate the references for template matching.

Two flow charts of the modified K-means clustering algorithm are given in Figure 5.1(a) and (b). Assuming that we are given a set of N observations $S = \{x_1, x_2, \ldots, x_N\}$ where every observation is either the whole or part of a pattern representing a replication of a specific spoken word. In our case, the pattern is the collection of the six ETP vectors corresponding to the temporal variation of the energy at six different frequency bands. Since it is intended that the clustering of the observations be based entirely on distance data, a matrix of distance D is defined as

$$D = \{d(x_i, x_j)\} \qquad i,j = 1, 2, \ldots, N \tag{5.1}$$

where $d(x_i, x_j)$ is the distance between the observations x_i and x_j as defined in (4.9). The minimax centre is found and is regarded as the cluster center of the entire training set. The minimax centre is defined as the pattern whose maximum distance to any other patterns within a particular cluster is minimum., i.e.,

$$c(S) = x_m \in S$$
$$\text{such that } \mathrm{Max}_j[d(x_m, x_j)] = \mathrm{Min}_i\{\mathrm{Max}_j[d(x_i, x_j)]\} \tag{5.2}$$

where $\mathrm{Max}_j[d(x_m,x_j)]$ denotes the maximum distance of pattern x_m to all other patterns x_j. The intracluster distance is computed by

$$D_{ic} = \mathrm{Max}_i[d(c(S_k), x_i)] \qquad i \in S_k \tag{5.3}$$

where $c(S_k)$ is the center of the cluster S_k and N_k is the number of patterns in the cluster S_k. There are two criteria which decide when

the procedure should stop. Firstly, we check whether the intracluster distances of all the created clusters have exceeded a predefined threshold, D_T. The procedure will continue iteratively until all intracluster distances are less than D_T. Secondly, we check whether the number of clusters (c_n) has already reached the desired number. In other words, one may create a fixed number of clusters or, alternatively, the number of clusters depends on the intracluster distances of the clusters which gives an indication of the variability of the reference tokens.

If neither of these conditions are satisfied, the clustering algorithm will proceed. In this case, the cluster with the largest D_{ic} is retrieved and is split up into two by assigning the two patterns x_a and x_b, whose distance between them, $d(x_a, x_b)$, is maximum within this cluster, as the new cluster centres. Now the number of cluster centres is incremented by one. Each pattern in the set S is then reclassified and reallocated to one of the clusters by choosing the one whose distance measure between the pattern and cluster centre is the minimum. After merging and splitting are completed, the minimax centre of each resulting cluster is again determined and is renamed as the new cluster center. A convergence check is made to see if any cluster has changed from the previous iteration. If it is so, the two preceding steps, i.e., cluster classifying and relocating of centre, will be repeated until no change in the clusters is observed. All the intracluster distances are then computed and the whole procedure will reiterate until the conditions on D_T or c_n are satisfied. Finally, we average all the patterns within each cluster to form a template for this particular data set S and the training procedure then stops.

In our tests, we first used the whole ETP matrices of the training tokens as the parameter set for clustering. However, when operating with such a large dimension of 6x16, time to time fluctuations might introduce excessive noise during the clustering process. This might cause adverse effect on the resulting templates providing poor representations of the vocabulary and possibly a reduction of recognition rate. In order to minimize this effect, the dimension of the parameter set was cut down by partitioning the ETP matrices along the time and frequency axes. Two kinds of parameter vectors were generated in this way. One of them was the ETP vectors of a particular frequency band while the other was the energy vectors at a particular time [30]. The dimensions of these vectors were 1x16 and 1x6 respectively. By using these parameter sets for clustering, we might, at the same time, observe the relations of variations among the segment energy along both the time and frequency axes. It was found that the vector sets formed less clusters than the matrix set for the same D_T, which implies that clustering has been done more effectively on the vectors. In order to cut down the dimension to a minimum, the individual segment energy, which was actually a scalar, was also used to form the parameter set for clustering. In this case, the number of clusters was further reduced. Recognition has been performed by using these four sets of reference templates and different results were obtained. In addition, the various effects of employing fixed and variable number of templates are observed as well. In the next section, all the tests will be delineated in detail and their results will also be given.

5.2 System Evaluation

The speech recognition system as described in the previous chapter has been implemented on a 16-bit microcomputer, IBM PC/XT, with 512k RAM. The software programs were written in PASCAL and recognition was performed off-line to facilitate easy implementation. The voice input was from a close-talking microphone to a cassette recorder whose amplifier has a pre-emphasis response. The recording was carried out in a laboratory with acoustic screening for echo suppression. The environment was in general quiet with only little ambient noise. The recorded utterances were later bandpass filtered at 100-3.3kHz, digitized with a 12-bit Analog-to-Digital converter at 8kHz sampling rate and stored in the hard disk of the microcomputer.

The recognition algorithm was initially tested by fifteen speakers, eight females and seven males, with a vocabulary consisting the ten Cantonese digits (1 - 10). Each speaker was requested to utter the ten isolated words twelve times, i.e., a total of 120 tokens, in which 20 of them (two for each digit) were used for training and the remaining 100 tokens were taken for testing. Therefore the sample size of the recognition test was totally 1500. The performance of the system was evaluated for both speaker-dependent and speaker-independent recognition. For speaker-independent recognition, a "semi-open test" as well as an "open test" were conducted. In the semi-open test, the same group of talkers provide data for both training and testing. Whereas in the open test, ten new speakers, three females and seven males, were asked to give a total of 500 tokens for testing. Various tests have been performed and the

results will be presented separately in the following sub-sections.

5.2.1 Speaker-Dependent Mode

In the speaker-dependent mode, the input tokens of each talker would be tested with the training data provided by himself. As mentioned earlier, each talker has been requested to provide two tokens for creating the reference templates. Therefore, when applying the clustering method, two occasions will occur. One occasion being that the two training tokens form two separate templates if the distance between them is larger than a threshold. Or, alternatively, they might combine together by simple averaging to form one single template if they are close to each other. The reference templates were created in three different ways. First of all, clustering was performed on the entire ETP matrix, thus, each word was represented by either one or two ETP patterns and we denote this method by M1. The second and the third method, on the other hand, were generated by applying clustering on ETP vectors of individual frequency bands and also on each single energy element. They are denoted by M2 and M3 respectively. Therefore, for M2, a spoken word might be represented by two ETP vectors for one frequency band and only one vector for the other. Similarly, for M3, there might be one template representing the energy at a particular segment in one band but two templates for the next segment in the same band or in the other band.

During template matching, if two templates exist in any instance, the distance between the parameter set of a test token and the reference word w was determined by taking the minimum of the two

distances between the test token and the two templates correspondingly. Hence, for M1, the distance measure is given by

$$D(w)[Sqr] = \mathrm{Min}_c \left\{ \sum_{q=1}^{6} \sum_{k=1}^{16} \frac{[\overline{E_q}(k) - R_{wc}\overline{E_q}(k)]^2}{\overline{E_q}(k) + R_{wc}\overline{E_q}(k)} \right\}$$

where Min_c denotes the minimum choice from the two clustered templates. Similarly, for M2, we have

$$D(w)[Sqr] = \sum_{q=1}^{6} \mathrm{Min}_c \left\{ \sum_{k=1}^{16} \frac{[\overline{E_q}(k) - R_{wc}\overline{E_q}(k)]^2}{\overline{E_q}(k) + R_{wc}\overline{E_q}(k)} \right\}$$

and also, for M3,

$$D(w)[Sqr] = \sum_{q=1}^{6} \sum_{k=1}^{16} \mathrm{Min}_c \left\{ \frac{[\overline{E_q}(k) - R_{wc}\overline{E_q}(k)]^2}{\overline{E_q}(k) + R_{wc}\overline{E_q}(k)} \right\}$$

where $c = 1$ or 2 which signifies the two templates of the word w. A rejection threshold was introduced in these tests such that an input utterance would be rejected if the difference between the smallest and next smallest distances corresponding to two different words was less than the threshold. For reliable and robust recognition, it is desirable to impose stricter rejection threshold. The one used in our tests was set to be 10 percent of the minimum distance, $\mathrm{Min}_w(D(w))$, such that the rejection criterion depended on the likeness of the test token and the reference set. The recognition decision for the speaker-dependent mode can therefore be stated as follow,

If [D(w2) - D(w1)] < 0.1 x D(w1) (5.4)

then REJECT

else w1 = RECOGNIZED digit

where D(w2) and D(w1) are the distances between the test token and the reference words w2 and w1, while D(w1) is the minimum among the whole vocabulary and D(w2) is the next minimum.

Apart from examining the three methods which use different parameter sets for clustering, experiments have also been performed to compare the recognition accuracy of the system when using different distance measures, namely, the squared distance, D(w)[Sqr], and the absolute distance, D(w)[Abs]. In addition, the performance of the system with and without classification was also recorded. Consequently, there were altogether twelve different combinations and hence twelve sets of results. These results are summarized in Table 5.1. A careful examination reveals that the recognition performance of using square distance measure and absolute distance measure has very little difference. Therefore, only the confusion matrices for D(j)[Sqr] are presented in Table 5.2(a) - 5.2(f) for the purpose of clarity.

5.2.2 Speaker-Independent Mode

For speaker-independent recognition, the reference set should consist of templates that are generated from the training data of many speakers. The training tokens of all the fifteen speakers were taken to form the training set. In our system, since there were two tokens for each digit from one speaker, a total of 30 tokens were available

for the creation of reference templates of a particular word. Both the semi-open test and the open test were carried out. For the semi-open test, the test data was taken from the fifteen trained speakers, making a sample size of 1500. Whilst in the open test, ten new speakers, three females and seven males, who had not participated in training, were invited to utter five times for each digit, thus providing altogether 500 test tokens. In addition to the three tests for the clustering algorithm in the speaker-dependent mode, a fourth trial (M4) was also performed in which the 1x6 vectors of energy in all frequency bands at a time segment were used as the parameter set for template generation. Two sets of templates were created for these four clustering methods. In the first set (T1), the number of clusters was fixed at eight so that the performances of the four methods of clustering in terms of recognition accuracies were compared under the same condition that the size of the reference sets were identical. In the second set (T2), the number of clusters was limited between a lower bound of two and an upper bound of eight, and within these limits, the actual number was determined by the intracluster distance as discussed in the previous section. A reduction of around 20 percent in the number of templates was found. The purpose of this experiment was to examine the recognition performance when the number of templates were reduced.

The decision scheme for the speaker-independent recognition was more sophisticated than the one used for speaker-dependent recognition in order to achieve a better performance. A two-pass decision based on the K-nearest neighbour (KNN) rule is used in which the vocabulary item whose average distance of the K-nearest neighbours to the unknown

utterance is minimum is chosen as the recognized word. Let $D(w)_n$ be the distance between a test token and the n^{th} nearest template of any digit w, then

$$D(w)_1 < D(w)_2 < D(w)_3 < D(w)_4 < D(w)_5 < D(w)_6 < D(w)_7 < D(w)_8$$

The task of the first-pass is to determine whether there is more than one vocabulary word which is acoustically similar to the test token. In this case, we set K=1 and check if the differences between the minimum distance and the others exceed a pre-defined threshold. If not, we move on to the second-pass to resolve these confusions. In our experiments, the threshold was set to 30 percent of the minimum distance. That is,

$$\text{If} \quad [D(w2)_1 - D(w1)_1) < 0.3 \times D(w1)_1 \qquad (5.5)$$

then proceed to SECOND PASS

else w1 = RECOGNIZED digit

where $D(w2)_1$ and $D(w1)_1$ are the distances between the test token and the reference words w2 and w1 such that $D(w1)_1$ is the minimum among the whole vocabulary. There might exist more than one $D(w2)_1$ for any digit $w2 \neq w1$ that satisfied the condition in (5.5). In the second-pass decision, we perform the KNN rule with K=2 and find the recognized word. A rejection criterion was imposed just as the same as in (5.4), i.e.,

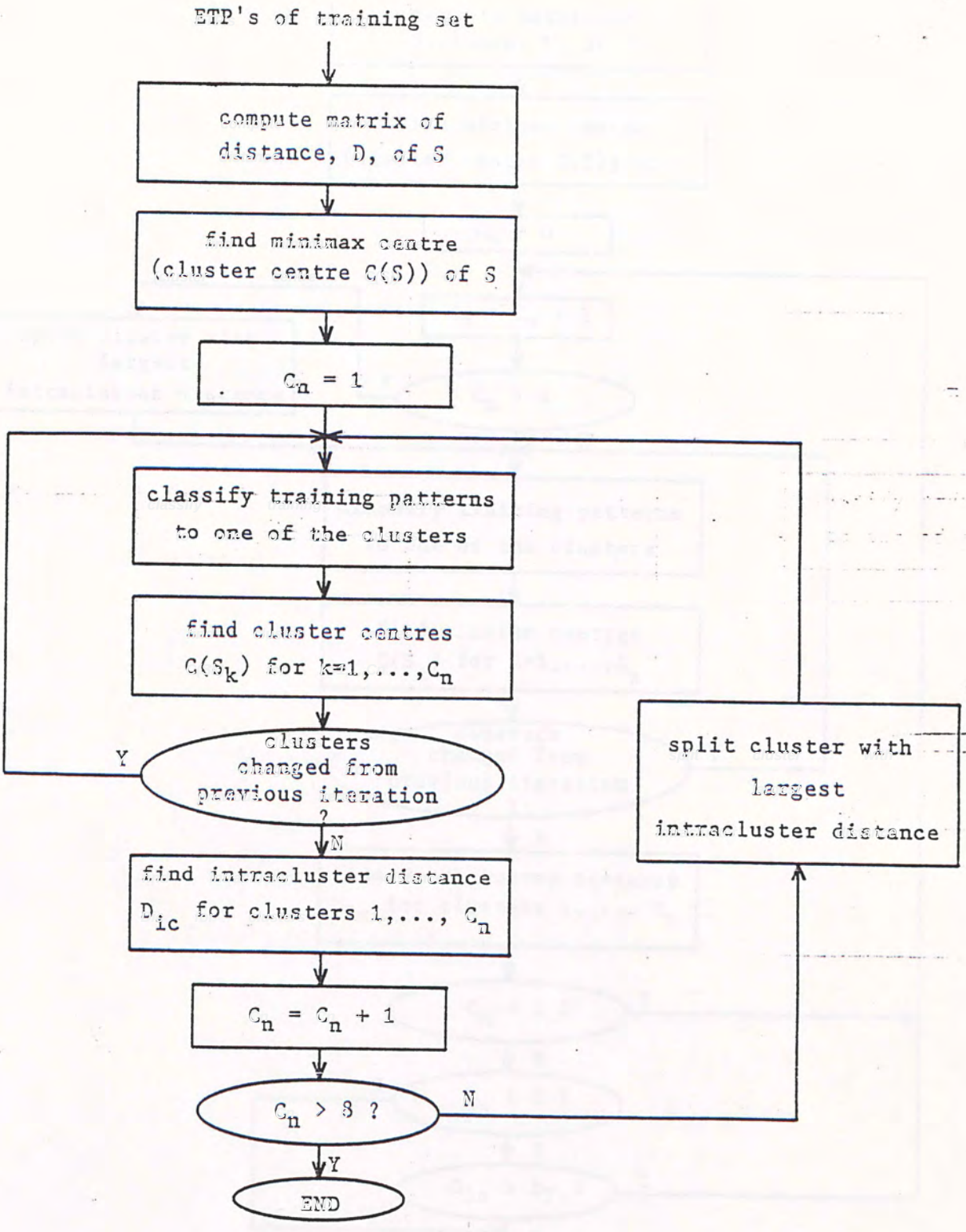
$$\text{If} \quad [(D(w2')_1 + D(w2')_2) - (D(w1')_1 + D(w1')_2)] < 0.1 \times (D(w1')_1 + D(w1')_2) \qquad (5.6)$$

then REJECT

else w1' = RECOGNIZED digit

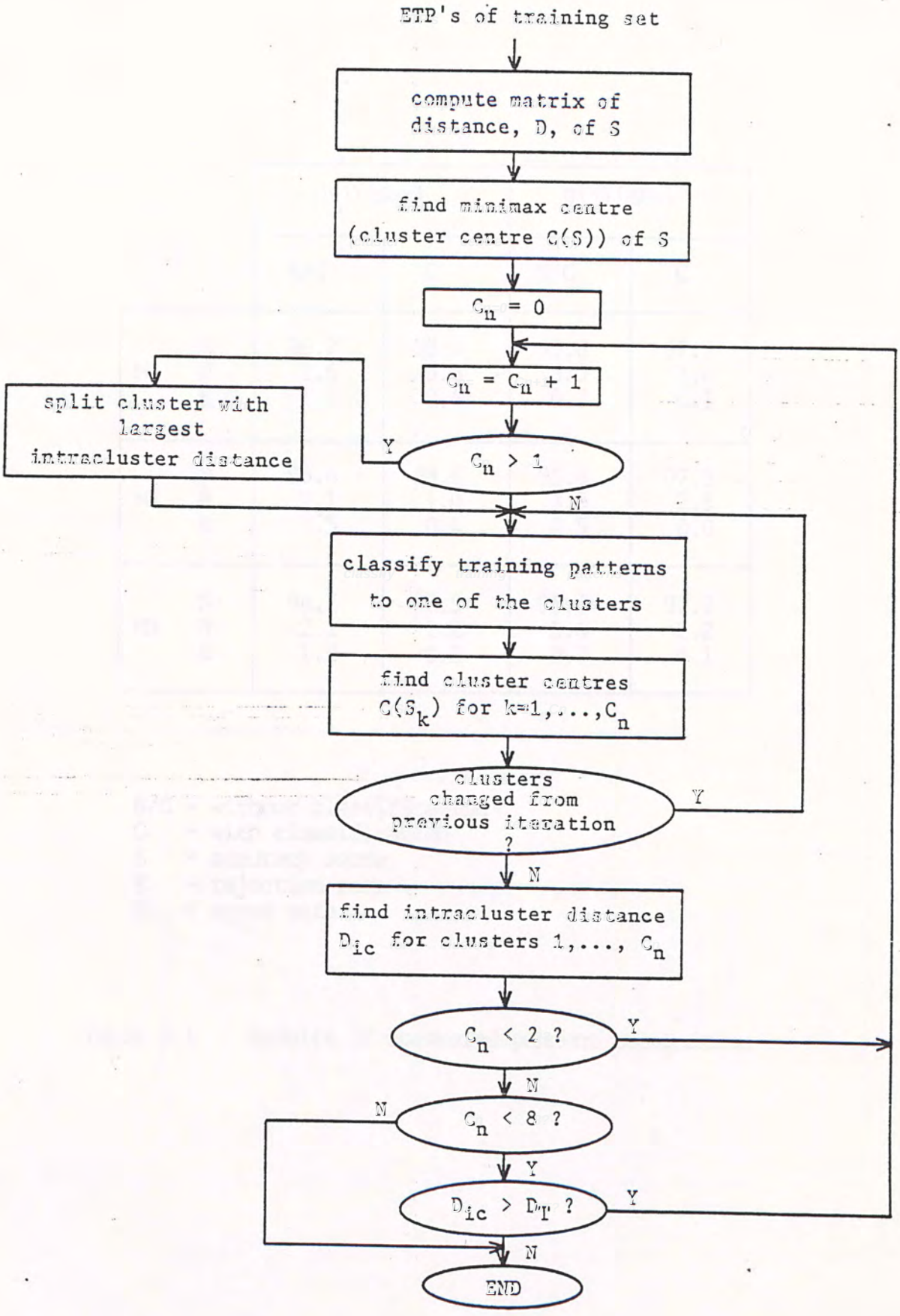
The classification procedure and the squared distance measure were utilized in all the tests for speaker-independent recognition. Since the four methods of clustering M1, M2, M3, and M4 were tested with two kinds of reference set, T1 and T2, there were totally eight experiments for each of the semi-open test and open test. Accordingly, 16 sets of results were obtained from all these trials and were summarized in Table 5.3, while Table 5.4 illustrates the confusion matrices of the semi-open test.

One thing we ought to mention is that due to the limitation of resources and manpower, we have only evaluated the performance of the proposed recognition scheme using a small vocabulary and very few number of speakers and tokens. However, the results show that this recognition scheme is fairly efficient and effective, and indeed, the performance would be improved if the sample size is increased. In the next chapter, we shall discuss in detail the results obtained in these experiments.



(a) Fixed number of clusters

Figure 5.1 Flow chart of the modified K-means clustering algorithm



(b) Number of clusters varies from 2 to 8

Figure 5.1 Flow chart of the modified K-means clustering algorithm

		D(j)[Sqr]		D(j)[Abs]	
		N/C	C	N/C	C
M1	S	96.7	99.0	95.8	97.5
	R	1.6	0.6	3.5	2.4
	E	1.7	0.4	0.7	0.1
M2	S	96.4	98.6	95.6	97.5
	R	2.1	1.0	3.9	2.5
	E	1.5	0.4	0.5	0.0
M3	S	96.7	98.5	95.9	97.7
	R	2.1	1.2	3.4	2.2
	E	1.2	0.3	0.7	0.1

N/C - without classification
C - with classification
S - accuracy score
R - rejection rate
E - error rate

Table 5.1 Results of speaker-dependent recognition

Test digit	Recognized as										Reject
	1	2	3	4	5	6	7	8	9	10	
1	133	0	3	0	0	0	5	0	0	0	9
2	0	137	0	9	0	0	0	0	0	0	4
3	0	0	149	0	0	0	0	0	0	0	1
4	0	0	0	149	0	0	0	0	0	0	1
5	0	0	0	0	150	0	0	0	0	0	0
6	1	0	1	0	0	146	0	0	0	2	0
7	1	0	0	0	0	0	145	0	0	0	4
8	0	0	1	0	0	0	0	148	0	0	1
9	0	0	0	0	0	0	0	0	150	0	0
10	0	0	0	0	0	0	3	0	0	143	4

Table 5.2(a) M1, without classification

Test digit	Recognized as										Reject
	1	2	3	4	5	6	7	8	9	10	
1	149	0	0	0	0	0	0	0	0	0	1
2	0	148	0	0	1	0	0	0	0	0	1
3	0	0	150	0	0	0	0	0	0	0	0
4	0	0	0	150	0	0	0	0	0	0	0
5	0	0	0	0	150	0	0	0	0	0	0
6	1	0	0	0	0	149	0	0	0	0	0
7	0	0	0	0	0	0	147	0	0	0	3
8	0	0	0	0	0	1	0	149	0	0	0
9	0	0	0	0	0	0	0	0	150	0	0
10	0	0	0	0	0	0	3	0	0	143	4

Table 5.2(b) M1, with classification

Test digit	Recognized as										Reject
	1	2	3	4	5	6	7	8	9	10	
1	133	0	3	0	0	0	1	0	0	0	13
2	0	136	0	8	0	0	0	0	0	0	6
3	0	0	150	0	0	0	0	0	0	0	0
4	0	0	0	149	0	0	0	0	0	0	1
5	0	0	0	0	150	0	0	0	0	0	0
6	1	0	1	0	0	145	0	0	0	2	1
7	2	0	0	0	0	0	142	0	0	0	6
8	0	0	2	0	0	0	0	148	0	0	0
9	0	0	0	0	0	0	0	0	150	0	0
10	0	0	0	0	0	0	3	0	0	143	4

Table 5.2(c) M2, without classification

Test digit	Recognized as										Reject
	1	2	3	4	5	6	7	8	9	10	
1	148	0	0	0	0	0	0	0	0	0	2
2	0	148	0	0	1	0	0	0	0	0	1
3	0	0	150	0	0	0	0	0	0	0	0
4	0	0	0	150	0	0	0	0	0	0	0
5	0	0	0	0	150	0	0	0	0	0	0
6	1	0	0	0	0	147	0	0	0	0	2
7	0	0	0	0	0	0	145	0	0	0	5
8	0	0	0	0	0	1	0	149	0	0	0
9	0	0	0	0	0	0	0	0	150	0	0
10	0	0	0	0	0	0	3	0	0	142	5

Table 5.2(d) M2, with classification

Test digit	Recognized as										Reject
	1	2	3	4	5	6	7	8	9	10	
1	144	0	1	0	0	0	1	0	0	0	4
2	0	141	0	8	0	0	0	0	0	0	1
3	0	0	150	0	0	0	0	0	0	0	0
4	0	0	0	148	0	0	0	0	0	0	2
5	0	0	0	0	149	0	0	0	0	0	1
6	1	0	0	0	0	145	0	0	0	1	3
7	2	0	0	0	0	0	132	0	0	2	14
8	0	0	2	0	0	0	0	148	0	0	0
9	0	0	0	0	0	0	0	0	150	0	0
10	0	0	0	0	0	0	0	0	0	143	7

Table 5.2(e) M3, without classification

Test digit	Recognized as										Reject
	1	2	3	4	5	6	7	8	9	10	
1	149	0	0	0	0	0	0	0	0	0	1
2	0	149	0	0	0	0	0	0	0	0	1
3	0	0	150	0	0	0	0	0	0	0	0
4	0	0	0	150	0	0	0	0	0	0	0
5	0	0	0	0	149	0	0	0	0	0	1
6	1	0	0	0	0	148	0	0	0	0	1
7	0	0	0	0	0	0	140	0	0	2	8
8	0	0	0	0	0	1	0	149	0	0	0
9	0	0	0	0	0	0	0	0	150	0	0
10	0	0	0	0	0	0	1	0	0	143	6

Table 5.2(f) M3, with classification

		Semi-open test		open test	
		T1	T2	T1	T2
M1	S	95.2	93.8	84.2	83.8
	R	2.4	2.1	5.4	6.4
	E	2.4	4.1	10.4	9.8
M2	S	93.6	92.9	83.4	82.4
	R	4.3	4.3	9.4	10.2
	E	2.1	2.8	7.2	7.4
M3	S	92.3	92.9	86.2	87.0
	R	3.7	3.8	7.0	6.4
	E	4.0	3.3	6.8	6.6
M4	S	95.2	95.3	85.2	86.2
	R	2.7	3.2	6.6	7.4
	E	2.1	1.5	8.2	6.4

S - accuracy score
R - rejection rate
E - error rate

Table 5.3 Results of speaker-independent recognition

Test digit	Recognized as										Reject
	1	2	3	4	5	6	7	8	9	10	
1	135	0	0	0	0	6	0	0	0	1	8
2	0	149	0	0	0	0	0	0	0	0	1
3	0	0	146	0	0	0	0	0	0	1	3
4	0	0	0	150	0	0	0	0	0	0	0
5	0	9	0	0	137	0	0	0	0	0	4
6	0	0	0	0	0	135	0	0	11	0	4
7	0	0	1	0	0	0	138	0	0	3	8
8	0	0	0	0	0	0	0	149	1	0	0
9	0	0	0	0	0	0	0	0	150	0	0
10	0	0	0	0	0	0	3	0	0	139	8

Table 5.4(a) M1, T1

Test digit	Recognized as										Reject
	1	2	3	4	5	6	7	8	9	10	
1	135	0	0	0	0	9	2	0	0	1	3
2	0	139	0	0	6	0	0	0	0	0	5
3	0	0	140	0	0	1	0	0	1	5	3
4	0	0	0	150	0	0	0	0	0	0	0
5	0	13	0	0	132	0	0	0	0	0	5
6	0	0	0	0	0	137	0	0	11	1	1
7	0	0	2	0	0	0	137	0	0	2	9
8	0	0	0	0	0	0	0	149	1	0	0
9	0	0	0	0	0	0	0	0	150	0	0
10	0	0	2	0	0	0	3	0	0	139	6

Table 5.4(b) M1, T2

Test digit	Recognized as										Reject
	1	2	3	4	5	6	7	8	9	10	
1	134	0	0	0	0	7	2	0	0	0	7
2	0	142	0	0	2	0	0	0	0	0	6
3	0	0	145	0	0	0	0	0	0	1	4
4	0	0	0	150	0	0	0	0	0	0	0
5	1	5	0	0	139	0	0	0	0	0	5
6	3	2	0	0	0	131	0	0	2	1	11
7	1	0	3	0	0	0	127	0	0	2	17
8	0	0	0	0	0	0	0	149	0	0	1
9	0	0	0	0	0	0	0	0	150	0	0
10	0	0	0	0	0	0	0	0	0	137	13

Table 5.4(c) M2, T1

Test digit	Recognized as										Reject
	1	2	3	4	5	6	7	8	9	10	
1	137	0	0	0	0	5	1	0	0	0	7
2	0	142	0	3	1	0	0	0	0	0	4
3	0	0	146	0	0	0	0	0	0	1	3
4	0	0	0	150	0	0	0	0	0	0	0
5	1	6	0	0	140	0	0	0	0	0	3
6	3	0	4	0	0	124	0	0	5	1	13
7	1	0	3	0	0	0	115	0	0	7	24
8	0	0	0	0	0	0	0	149	0	0	1
9	0	0	0	0	0	0	0	0	150	0	0
10	0	0	0	0	0	0	0	0	0	140	10

Table 5.4(d) M2, T2

Test digit	Recognized as										Reject
	1	2	3	4	5	6	7	8	9	10	
1	134	0	0	0	0	6	3	1	0	0	6
2	0	136	0	0	9	0	0	0	0	0	5
3	0	0	146	0	0	0	0	0	0	1	3
4	0	0	1	149	0	0	0	0	0	0	0
5	0	7	0	0	143	0	0	0	0	0	0
6	2	1	0	0	0	129	0	0	9	0	9
7	3	0	2	0	0	0	114	0	0	10	21
8	0	0	0	0	0	1	0	149	0	0	0
9	0	0	0	0	0	0	0	0	150	0	0
10	1	0	1	0	0	2	0	0	0	139	7

Table 5.4(e) M3, T1

Test digit	Recognized as										Reject
	1	2	3	4	5	6	7	8	9	10	
1	136	0	0	0	0	5	3	1	0	0	5
2	0	138	0	0	5	0	0	0	0	0	7
3	0	0	147	0	0	0	0	0	1	1	1
4	0	0	1	149	0	0	0	0	0	0	0
5	0	5	0	0	145	0	0	0	0	0	0
6	4	0	0	0	0	127	0	0	9	0	10
7	3	0	2	0	0	0	119	0	0	6	20
8	0	0	0	0	0	1	0	148	0	0	1
9	0	0	0	0	0	0	0	0	146	0	4
10	1	0	1	0	0	0	0	0	0	139	9

Table 5.4(f) M3, T2

Test digit	Recognized as										Reject
	1	2	3	4	5	6	7	8	9	10	
1	134	0	0	0	0	6	2	0	0	1	7
2	0	150	0	0	0	0	0	0	0	0	0
3	0	0	146	0	0	0	0	0	1	1	2
4	0	0	0	150	0	0	0	0	0	0	0
5	0	11	0	0	134	0	0	0	0	0	5
6	0	0	0	0	0	139	0	0	1	3	7
7	0	0	0	0	0	0	136	0	0	4	10
8	0	0	0	0	0	0	0	147	0	0	3
9	0	0	0	0	0	0	0	0	149	0	1
10	0	0	0	0	0	0	2	0	0	143	5

Table 5.4(g) M4, T1

Test digit	Recognized as										Reject
	1	2	3	4	5	6	7	8	9	10	
1	134	0	0	0	0	5	2	0	0	1	8
2	0	147	0	0	2	0	0	0	0	0	1
3	0	0	146	0	0	0	0	0	0	1	3
4	0	0	0	150	0	0	0	0	0	0	0
5	0	7	0	0	139	0	0	0	0	0	4
6	0	0	0	0	0	139	0	0	0	1	10
7	0	0	1	0	0	0	135	0	0	2	12
8	0	0	0	0	0	0	0	147	0	0	3
9	0	0	0	0	0	0	0	0	149	0	1
10	0	0	0	0	0	0	1	0	0	143	6

Table 5.4(h) M4, T2

CHAPTER 6 DISCUSSION AND CONCLUSION

A number of achievements have been accomplished by the proposed recognition system. The endpoint detection algorithm that has been used is simple and efficient and can be implemented very easily using either software or hardware. The beginning and end of all the utterances, including the training and testing tokens, had been correctly located. When this algorithm was utilized in a software environment, the speed was limited mainly by the calculation of the energy and zero-crossing contour. However, these parameters can be obtained easily using external integrators and voltage comparators thereby leaving only the decisions for the system processing unit, and hence the time needed for endpoint detection could be reduced considerably.

The accuracy of the classification algorithm was also impressive. For 1500 tokens uttered by 15 speakers, only 125 were unclassified and no one was identified to the wrong group. Thus there was a saving of approximately 43 percent in terms of the number of comparisons needed before a recognition decision could be made. Of course, this improvement inherently depends on the distribution of the number of templates for each member of the two groups and also depends on the vocabulary itself. In addition, the performance of the classification scheme is also an important factor. If a large proportion of the input tokens are unclassified, then the effectiveness of reducing computation will be greatly lessened.

By comparing the results shown in Table 5.1, we can see that the recognition accuracy was higher if classification was performed prior

to template matching. In fact, an average of 2 percent improvement was found in all methods M1, M2 and M3 for both D(w)[Sqr] and D(w)[Abs], in which one percent corresponded to the drop in rejection rate while the remaining one percent came from the reduction in error rate. This observation can be further confirmed by investigating the confusion matrices. When the classification process was eliminated, a relatively large proportion of the digit "1" was recognized as "3" or "7", and the digit "7" was also confused with "1". This is because within our selected vocabulary, that is, the ten Cantonese digits, some are having very similar or even the same vowel, such as the digits "1", "3", "7" and "10", while some are extremely different, such as the digits "4", "8" and "9". Besides, one should notice that many of the digit "2", which came from one particular speaker, was recognized as "4". Since "2" and "4" are having completely different in phonetic labelling, this phenomenon may be regarded as the pronounciation characteristic of that specific speaker. However, if classification was made in the process, the digit "1", after assigning to the VI group, would not be compared with digits "3" and "7" which belong to the FI group, and similarly for the reverse case as well as for "2" and "4". Hence the confusions were avoided and a significant increase could be found in the accuracy of recognizing the digits "1", "2" and "7". Therefore, the classification strategy improves both the recognition speed as well as the overall recognition rate.

Although the classification algorithm has achieved high performance, its implementation was not at all difficult. The decisions were straight forward and software realization is easy. This simplicity resulted in a fast preliminary class decision with little

time consumption.

During feature extraction, five digital filters were executed in sequence before the ETP vectors were computed. In order to achieve the specific bandpass characteristics, fairly high order digital filters had been used. Therefore a great deal of multiplications were required and much time was spent in the filtering process. This has slowed down the recognition speed to a great extent. However, if the bandpass filtering was performed using analog filters, the computational load would be reduced and almost real-time processing is possible. In that case, instead of filtering the digitized signal, the input speech is first sent through the analog filter bank before analog-to-digital conversion. These five output signals together with the original input are then sampled individually to form six sequences. Thus, the only job of the system processing unit is to calculate the ETP vectors from these sequences and then perform the recognition.

Since the dimension of the feature vector was fixed for any input utterance, the dynamic time warping procedure could be eliminated when calculating the distance between a test input and reference templates. This results in a fast and simple comparison strategy using the Euclidean distance measure and self-normalization. Of course, this linear time alignment method is not as good as the conventional DTW algorithm, but it is sufficient when dealing with monosyllabic words only. In addition, a non-linear normalization technique is adopted to effectively equalize the degree of importance of large and small energy magnitudes. It has been shown that the number of confusions

between digits having the same vowel increased significantly when no normalization was performed. Both D(w)[Sqr] and D(w)[Abs] have achieved high recognition accuracies in either method of clustering, M1, M2 or M3. Since the squaring operation was omitted, the use of the absolute distance measure, D(w)[Abs], can speed up the recognition procedure, but with a penalty of a drop of one percent in the overall accuracy. On the other hand, the error rates of all three methods were reduced to a minimum when using D(w)[Abs]. Therefore, for any particular applications, if speed is critical and rejection is tolerable, the absolute distance measure is highly recommended.

The rejection criterion introduced in the decision process was to reduce incorrect recognition arising from confusions between similar digits such as "7" and "10". In our tests, all the test tokens were members of the selected vocabulary. But for a real application, this assumption will no longer hold. Accordingly, the system should be able to screen out those input that do not belong to the vocabulary. This may be achieved by checking the distance between the ETP vectors of the input token and the reference templates. If the distance D(w) is large, it is sure that the input is not the word w. Hence a word should be rejected if the minimum of the distances exceeds a predefined threshold, which means that this word is very different from all the words in the reference set.

For speaker-dependent recognition, it was found that M1 performed slightly better than the other two clustering methods in terms of accuracy, when the squared distance D(w)[Sqr] was used. But if the absolute distance D(w)[Abs] was used instead, there was scarcely any difference between the performance of the three clustering methods.

In addition, one might notice that for both squared and absolute distance mearures, the error rates of the three method were almost identical. In other words, the lower accuracy is due to an increase of rejection rate only. On the other hand, the memory requirement of these methods were quite different. Clustering based on the whole ETP matrices consumed most memories. Almost all the words required two templates for representation. Whilst clustering based on segment energy had the least number of templates because most of the segments were represented by one value only. The ratio of memory requirement of these three methods M1:M2:M3 was about 1:0.83:0.7. That means, for a drop of not more than 0.5 percent in accuracy, M2 and M3 have saved about 17 and 30 percent of memory respectively. This saving also resulted in a shortening of recognition time since the number of comparisons was reduced due to less templates. Therefore, M3 is probably the best choice for speaker-dependent applications.

In the semi-open test of speaker-independent recognition, the mean accuracy for T1 was 95.2, 93.6, 92.3 and 95.2 percent for M1, M2, M3 and M4 with a rejection rate of 2.4, 4.3, 3.6 and 2.7 percent respectively. The results of M1 and M4 are comparable to each other and are very satisfactory while for M2 and M3, the results are slightly worse. That means for the same number of templates, M1 and M4 give the best performance. For T2, when the total number of templates used in the reference set were reduced by 20 percent, we found that there was only a marginal decrease in the overall recognition accuracy for M2, and even a slight increase for M3 and M4. However, for M1, a drop of roughly 1.4 percent in accuracy was recorded. This shows that the recognition score is dependent on the

number of templates in the reference set when the templates were being generated by clustering whole ETP matrices. In the open test, the recognition score dropped significantly. The low accuracy implied that the reference set is unable to represent all the variabilities in saying words by different talkers. In fact, using only 15 speakers to form the training set is hardly enough to give good templates for speaker-independent recognition. One of the ten untrained speakers achieved a recognition accuracy of as low as 62 percent with the digits "5", "7" and "10" completely mis-recognized to other words. This confirms that the reference set was indeed poorly represented. More rigorous tests using a large population for training is in progress and better results are expected. Although the results were preliminary, we have noticed that the recognition accuracy of M3 and M4 was significantly higher than the other two methods for both T1 and T2. Besides, with reduced templates, M3 and M4 gave better results. It seems that the templates will be able to represent more variabilities if clustering is done on a parameter set with smaller dimension. Since the sample size for testing is not adequate, we cannot draw any definite conclusion, but obviously M4 has shown an outstanding performance over the other methods.

Although the proposed isolated-word recognition system has shown good performance, there is much room for further studies. Though this system was evaluated using Cantonese only, it may be applied to other monosyllabic languages, such as Mandarin and other Chinese dialects. Besides, the vocabulary selected for testing is too small for actual applications. Some command words for robotic control, for example, may be added to simulate a more realistic environment. But as the

vocabulary size grows, the time required for distance measure will increase accordingly. This undesirable phenomenon may be avoided by a more sophisticated classification scheme which divides the whole vocabulary into many small sub-groups. In this case, comparisons will only be done within each group and thus speeds up the recognition process. Apart from grouping according to the initials, we may also classify the finals. For Cantonese, the finals can be phonetically realized as either nasal, vowel-like or stop. Then combining the initial classification, six groups may be created. Of course, the actual development of the algorithm requires detailed analysis of the finals' temporal as well as spectral characteristics.

There is no definite rule governing the dimension of the ETP vectors, that is, the number of segments for energy measurement can in fact be varied. The choice of 16 segments is strictly on an ad hoc basis. If more segments are used, the effects of time-warping will become more pronounced, and if the block length becomes too long, that is, the number of blocks become less, the stationarity of the speech signal will no longer be valid. However, if there are too many segments, the computational load will be increased. Hence a compromise between these two factors must be made. Experiments may be performed using different vector sizes so as to find the optimum choice.

As a conclusion, a novel isolated-word recognition system is proposed which is suitable for both speaker-dependent and speaker-independent applications. Although the results obtained are preliminary due to the limited vocabulary and speakers being used,

especially for the speaker-independent recognition, they have shown that the methodology is sound. One of the major achievement of this system is the elimination of the time-warping process by using the ETP vectors of a word as the parameter for recognition. Because the speech signals are effectively linearly aligned, it is particularly suitable for monosyllabic languages. The algorithms that are employed are simple and can be easily implemented on a microcomputer. As described previously, a large part of the pre-processing task can be realized by standard hardware, thus a real-time recognition system may be achieved cheaply. This enables a wide range of practical applications such as aids for the disabled, automatic voice-controlled cashier, simple robotic control, etc.

REFERENCES

[1] J.L.Flanagan, "Computers that talk and listen : man-machine communication by voice", Proc. IEEE, Vol. 64, No. 4, pp.405-415, April 1976.

[2] J.D.Markel and A.H.Gray, "Linear Prediction of Speech", Springer Verlag.

[3] A.Ichikawa, Y.Nakano, and K.Nakata, "Evaluation of various parameter sets in spoken digits recognition", IEEE Tran. on Audio and Electroacoustics, Vol. AU-21, No. 3, pp.202-209, June 1973.

[4] F.Itakura, "Minimum prediction residual principle applied to speech recognition", IEEE Tran. on Acoustics, Speech, and Signal Processing, Vol. ASSP-23, No. 1, pp.67-72, February 1975.

[5] M.K.Brown and L.R.Rabiner, "On the use of energy in LPC-based recognition of isolated words", The Bell System Technical Journal, Vol. 61, No. 10, pp.2971-2987, December 1982.

[6] G.M.White and R.B.Neely, "Speech recognition experiments with linear prediction, bandpass filtering, and dynamic programming", IEEE Tran. on Acoustics, Speech, and Signal Processing, Vol. ASSP-24, No.2, pp.183-188, April 1976.

[7] M.R.Sambur and L.R.Rabiner, "A speaker-independent digit-recognition system", The Bell System Technical Journal, Vol. 54, No. 1, pp.81-102, January 1975.

[8] G.J.Vysotsky, "A speaker-independent discrete utterance

recognition system, combining deterministic and probabilistic strategies", IEEE Tran. on Acoustics, Speech, and Signal Processing, Vol. ASSP-32, No. 3, pp.489-499, June 1984.

[9] G.J.Vysotsky, "Speaker-independent isolated word recognition using a one-pass analysis", ICASSP, Vol. 1, pp. 9.10.1-4, 1984.

[10] G.E.Kopec and M.A.Bush, "Network-based isolated digit recognition using vector quantization", IEEE Tran. on Acoustics, Speech, and Signal Processing, Vol. ASSP-33, No. 4, pp.850-867, August 1985.

[11] L.R.Rabiner, S.E.Levinson, and M.M.Sondhi, "On the application of vector quantizatin and hidder Markov models to speaker-independent, isolated word recognition", The Bell System Technical Journal, Vol. 62, No. 4, pp.1075-1105, April 1983.

[12] B.H.Juang, L.R.Rabiner, "Mixture autoregressive hidden Markov models for speech signals", IEEE Tran. on Acoustics, Speech, and Signal Processing, Vol. ASSP-33, No. 6, pp.1404-1413, December 1985.

[13] D.Bursky, "Speech recognition builds its vocabulary to handle more tasks", Electronic Design, pp.113-124, April 18, 1985.

[14] J.G.Wilpon, L.R.Rabiner, "A modified k-means clustering algorithm for use in isolated word recognition", IEEE Tran. on Acoustics, Speech, and Signal Processing, Vol. ASSP-33, No. 3, pp.587-594, June 1985.

[15] W.A.Ainsworth, "Mechanisms of speech recognition", Pergamon Press Ltd., 1976.

[16] L.R.Rabiner and R.W.Schafer, "Digital processing of speech signals", Prentice-Hall, Englewood Cliffs, New Jersey, 1987.

[17] A.H.Gray and J.D.Markel, "Distance measures for speech processing", IEEE Tran. on Acoustics, Speech, and Signal Processing, Vol. ASSP-24, No. 5, pp.380-391, October 1976.

[18] B.A.Dautrich, L.R.Rabiner, and T.B.Martin, "On the effects of varying filter bank parameters on isolated word recognition", IEEE Tran. on Acoustics, Speech, and Signal Processing, Vol. ASSP-31, No. 4, pp.793-806, August 1983.

[19] S.Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum", IEEE Tran. on Acoustics, Speech, and Signal Processing, Vol. ASSP-34, No. 1, pp. 52-59, February 1986.

[20] H.Sakoe and S.Chiba, "Dynamic programming algorithm optimization for spoken word recognition", IEEE Tran. on Acoustics, Speech, and Signal Processing, Vol. ASSP-26, No. 1, pp.43-49, February 1978.

[21] H.Sakoe and S.Chiba, "A dynamic programming approach to continuous speech recognition", Proc. 7th Int. Congr. Acoust., Paper 20C-13, August 1971.

[22] C.Myers, L.R.Rabiner, and A.E.Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition", IEEE Tran. on Acoustics, Speech, and Signal Processing, Vol. ASSP-28, No. 6, pp.623-635, December 1980.

[23] D.K.Burton, J.E.Shore, and J.T.Buck, "Isolated-word speech recognition using multisection vector quantization codebooks", IEEE Tran. on Acoustics, Speech, and Signal Processing, Vol. ASSP-33, No. 4, pp.837-848, August 1985.

[24] L.F.Lamel, L.R.Rabiner, A.E.Rosenberg, and J.G.Wilpon, "An improved endpoint detector for isolated word recognition", IEEE Tran. on Acoustics, Speech, and Signal Processing, Vol. ASSP-29, No.4, pp.777-785, August 1981.

[25] W.M.Lai, P.C.Ching, and Y.T.Chan, "Isolated word recognition using energy-time profiles", Int. Journal of Electronics (to be published).

[26] S.L.Wong, "A Chinese syllabary pronounced according to the dialect of Canton", Chung Wah Book Store, Hong Kong, 1984.

[27] L.R.Rabiner, "On creating reference templates for speaker independent recognition of isolated words", IEEE Tran. on Acoustics, Speech, and Signal Processing, Vol. ASSP-26, No. 1, pp.34-42, Feburary 1978.

[28] S.E.Levinson., L.R.Rabiner, A.E.Rosenberg, and J.G.Wilpon, "Interactive clustering techniques for selection speaker-independent reference templates for isolated word recognition", IEEE Tran. on Acoustics, Speech, and Signal Processing, Vol. ASSP-27, No. 2, pp.134-141, April 1979.

[29] L.R.Rabiner, S.E.Levinson, A.E.Rosenberg, and J.G.Wilpon, "Speaker-independent recognition of isolated words using

clustering techniques", IEEE Tran. on Acoustics, Speech, and Signal Processing, Vol. ASSP-27, No. 4, pp.336-349, August 1979.

[30] P.C.Ching, W.M.Lai, and Y.T.Chan, "An efficient algorithm for isolated word recognition of monosyllabic languages", Proc. of European Conference on Speech Technology, Edinburgh, September 2-4, 1987.